



Carlos Eduardo Correia de Passos

Nº 44574

Elaboração de Horários Académicos

Dissertação de Mestrado Integrado em Engenharia Informática

2º Semestre, 2015/2016

Orientador: Pedro Barahona, Professor Doutor, UNL



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro, 2016

© Copyright

Carlos Eduardo Correia de Passos

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Todo o trabalho desenvolvido não seria possível sem a orientação, apoio e ajuda de várias pessoas. Desde já expresso toda a minha gratidão.

Gostaria de começar por enaltecer, reconhecer e agradecer toda a sábia orientação e ajuda que o Professor Doutor Pedro Barahora nunca deixou de humildemente me proporcionar. Terá para sempre o meu reconhecimento.

Não podia deixar de citar os meus verdadeiros pilares em todo o desenrolar do trabalho. A minha esposa, Vera, que sempre me apoiou e nos momentos cruciais se sacrificou ao acarretar todas as responsabilidades familiares, à minha filha, Joana, que soube sempre mostrar um sorriso quando eu precisava, aos meus pais e ao meu irmão que mesmo longe estiveram sempre perto de mim.

Finalizo com uma palavra de apreço à Força Aérea por me ter permitido e facilitado o acesso a informação pertinente para o desenvolvimento do trabalho.

Resumo

A geração de horários é uma tarefa de dificuldade elevada e requer trabalho árduo devido à necessidade de gerir os diversos conflitos de restrições impostos aos recursos a ser usados, mais concretamente alunos, professores e salas.

A grande variedade de restrições associada às diferentes necessidades de diferentes sistemas de ensino e até mesmo entre escolas do mesmo nível, tem dificultado a elaboração de formatos *standard* que permitam não só caracterizar as próprias restrições/regras mas também os recursos de grandes variedades de sistemas de ensino, possibilitando a comparação de práticas e avaliação de desempenho assim como proporcionar uma estrutura capaz de ser manipulada por sistemas com capacidade de gerar horários.

Esta dissertação aborda o problema da elaboração de horários académicos, focando-se no caso da Academia da Força Aérea (AFA). São apresentados vários exemplos de sistemas de ensino e um formato de especificação em XML para *benchmarking* de horários académicos, no qual são especificadas as restrições impostas aos horários da Academia da Força Aérea. Várias técnicas de pesquisa local restringida que são tradicionalmente usadas para resolver este tipo de problemas, nomeadamente as técnicas de *Hill-Climbing* (HC), *Simulated Annealing* (SA) e *Tabu Search* (TS), são discutidas e são exploradas para resolver o problema da geração de horários na AFA.

Este trabalho avalia esta abordagem e foi elaborada uma ferramenta para resolução de horários académicos, que para além de validar a completude de informação fornecida na representação XML (e estendê-la), permite obter soluções que satisfazem um conjunto de restrições obrigatórias (como a não sobreposição de recursos) e otimizam um conjunto de preferências adicionais (boas práticas pedagógicas, como a não existência de furos). A eficiência da ferramenta é estudada por comparação com ferramentas que já utilizam a representação XML referida.

Palavras-chave: Horários Académicos, *Benchmarking*, Formato em XML, Pesquisa Local, *Hill-Climbing*, *Simulated Annealing*, *Tabu Search*

Abstract

Timetabling generation is a task of high difficulty requiring hard work given the need to manage several conflicting constraints on resources to be used, namely students, teachers and classrooms.

The wide variety of restrictions associated with the different needs of different educational systems and even between schools of the same level, has hindered the development of standard formats that not only fully characterize restrictions/rules but also the resources of different education systems, enabling the comparison of practices and performance evaluation as well as providing a structure capable of being manipulated by systems capable of generating schedules.

This dissertation addresses the problem of academic timetabling generation, focusing on the case of the Air Force Academy (AFA). It presents several examples of school systems and an XML specification format for benchmarking of academic schedules, in which the schedules restrictions in Air Force Academy are specified. Several constrained local search techniques that have been traditionally used to solve this type of problems, namely Hill-climbing (HC), Simulated Annealing (SA) and Tabu search (TS), are discussed and are explored to solve the timetabling problem in AFA.

This work evaluate this approach and a tool was elaborated for implementing academic schedules, which in addition to validating the completeness of information provided in the XML representation (and extending it), allows for seeking solutions that satisfy a set of mandatory restrictions (such as no overlap of resources) and optimize a set of additional preferences (good teaching practices, such as the lack of idle times). The efficiency of the tool is studied by comparison with tools that already use the referred XML representation.

Keywords: Academic Timetable, Benchmarking, XML format, Local Search, Hill-Climbing, Simulated Annealing, Tabu Search

Índice

1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Descrição do Problema.....	1
1.3	Solução do Problema e Contribuições	2
1.3.1	Solução do Problema	2
1.3.2	Contribuições	2
2	Trabalho Relacionado	3
2.1	Problema de Horários Acadêmicos	3
2.1.1	Austrália	4
2.1.2	Reino Unido	5
2.1.3	Finlândia	5
2.1.4	Grécia	6
2.1.5	Holanda	7
2.1.6	Horários na AFA/FAP	7
2.2	Formato para benchmarking.....	8
2.2.1	Tempos letivos e recursos.....	9
2.2.2	Eventos.....	9
2.2.3	Regras	10
2.2.4	Formato XML	12
2.3	Pesquisa Local.....	14
2.3.1	Hill-Climbing	17
2.3.1.1	Late Acceptance Hill-Climbing	18
2.3.1.2	Step Counting Hill-Climbing	18

2.3.2	Simulated Annealing	19
2.3.3	Tabu Search.....	20
2.3.4	Outros	21
3	TimeTabling	22
3.1	Implementação MVC.....	23
3.1.1	Controller.....	24
3.1.2	Model.....	25
3.1.3	View	27
3.2	Plano do trabalho	34
3.3	Implementação da Pesquisa Local.....	34
3.3.1	Random-restart hill-climbing	34
3.3.2	Late Acceptance Hill-Climbing.....	35
3.3.3	Step Counting Hill-Climbing	36
3.3.4	Simulated Annealing	36
3.3.5	Tabu Search.....	37
3.4	Material utilizado.....	37
4	Resultados Experimentais	38
4.1	Baseadas em horários existentes.....	39
4.1.1	Alteração da tarde livre para outro dia	40
4.1.2	Indisponibilidade de dois professores para um dia	41
4.1.3	Indisponibilidade de uma sala toda a semana	41
4.2	Através de geração aleatória de horários	42
4.2.1	Horários para a AFA	42
4.2.2	Horários para benchmark de escola da Grécia	43
5	Conclusões.....	45
	Bibliografia.....	47

Webliografia	49
A. Apêndice	51
A.1 Exemplo de um formato XML para horários académicos da AFA.....	52
A.2 Exemplo num formato XML de restrições aos horários académicos da AFA.....	56
A.3 Resultados experimentais baseados no horário da AFA com a alteração da tarde livre para outro dia	61
A.4 Resultados experimentais baseados no horário da AFA com a indisponibilidade de dois professores.....	63
A.5 Resultados experimentais baseados no horário da AFA com a indisponibilidade de uma sala	65
A.6 Resultados experimentais através de geração aleatória de horários para a AFA ...	67
A.7 Resultados experimentais através de geração aleatória de horários para escola da Grécia	70

Ilustrações

Ilustração 1: Algoritmo de Pesquisa Local	15
Ilustração 2: Exemplo de Plateaus em Pesquisa Local	16
Ilustração 3: Modelo MVC	23
Ilustração 4: Código do TimeTabling	24
Ilustração 5: Entrada no sistema.....	28
Ilustração 6: Ecrã inicial do sistema.....	28
Ilustração 7: Upload de ficheiros	29
Ilustração 8: Seleção de ficheiros.....	29
Ilustração 9: Criação de ficheiros.....	29
Ilustração 10: Detalhe da instância.....	30
Ilustração 11: Continuação do detalhe da instância	31
Ilustração 12: Detalhe da solução.....	31
Ilustração 13: Horário.....	32
Ilustração 14: Gráficos	33
Ilustração 15: Geração automática	33
Ilustração 16: Resultados experimentais com a técnica Random-restart hill-climbing	39
Ilustração 17: Média dos resultados experimentais da alteração da tarde livre	40
Ilustração 18: Média dos resultados experimentais com a indisponibilidade de dois professores	41
Ilustração 19: Média dos resultados experimentais com a indisponibilidade de uma sala .	41
Ilustração 20: Média dos resultados experimentais da geração aleatória de horários para a AFA.....	42
Ilustração 21: Média dos resultados experimentais da geração aleatória de horários para a Grécia	43

Tabelas

Tabela 1: Parâmetros utilizados das técnicas de pesquisa local	38
--	----

1 Introdução

Todos os estabelecimentos de ensino se deparam no início de cada momento académico com a necessidade de criar horários académicos.

A criação de um horário académico é uma tarefa de difícil execução, implica a gestão de restrições entre alunos, professores e salas em determinados tempos letivos, e que deve ser suportada por sistemas de informação evoluídos.

A conjugação de sistemas capazes de gerir informação académica e técnicas de pesquisa e otimização eficientes, pode permitir a geração semiautomática de horários académicos com um desempenho que na grande maioria das vezes não seria capaz de ser atingido com um tratamento tradicional, essencialmente manual.

1.1 Contexto e Motivação

A Força Aérea Portuguesa (FAP) tem na Direção de Comunicações e Sistemas de Informação (DCSI), entre várias outras, a missão de desenvolver sistemas de informação. Um dos sistemas desenvolvidos é o sistema de gestão escolar (SGE) da Academia da Força Aérea (AFA). O SGE engloba tarefas inerentes à formação académica e gestão dos seus recursos.

Embora se tenha atingido um grau de maturidade elevado do sistema, este ainda tem capacidade para evoluir, e a motivação desta dissertação prendeu-se com uma necessidade sentida na FAP de geração automática dos horários académicos. Com o colmatar desta necessidade vai ser possível de uma maneira célere e eficiente a criação de variados horários académicos, evitando-se perdas de tempo nesta tarefa e assim poder disponibilizar recursos humanos a outro tipo de tarefas.

1.2 Descrição do Problema

No âmbito desta dissertação, como universo de estudo, são utilizadas 5 turmas, com uma média de 7 horas diárias de aulas divididas por 5 dias e cerca de 10 unidades curriculares, o que envolveu mais de 1750 variáveis. O tratamento manual deste universo associado com professoras e salas para a geração de horários académicos não é de todo aconselhável.

O objetivo é portanto colmatar a geração manual de horários académicos com a geração automática dos mesmos. A este problema acresce a necessidade de na geração automática saber gerir as restrições/regras impostas aos próprios recursos e tempos letivos e satisfazer variadas preferências de diferentes graus.

A automatização de procedimentos é concretizada utilizando técnicas eficientes, a resolução do atual problema passa por técnicas que proporcionam resultados eficientes e céleres, mas também otimizados.

1.3 Solução do Problema e Contribuições

1.3.1 Solução do Problema

A solução do problema passou pela implementação de um sistema para geração de horários académicos. Nesta implementação foi utilizada informação num formato em XML para *benchmarking* de horários académicos. A informação referente às turmas, aulas e tempos letivos é proveniente da estrutura de dados do sistema da Força Aérea, a restante informação foi gerada de modo a salvaguardar informação sensível. Esta informação caracteriza todos os recursos e foram adicionadas um conjunto de restrições obrigatórias e outras preferências.

1.3.2 Contribuições

Para além de um sistema de resolução de horários académicos, esta dissertação contribui para o uso de um formato em XML para *benchmarking* de horários académicos que funcione para um leque variado de sistemas de ensino e diferentes estabelecimentos de ensino. Outra contribuição é a implementação de técnicas de pesquisa local restringida para resolver o problema de obtenção de horários especificados nesse formato.

2 Trabalho Relacionado

Neste capítulo resume-se alguma bibliografia pesquisada no contexto da preparação da dissertação, focada em 3 áreas distintas para o desenvolvimento da dissertação:

1. Especificidade de horários em vários contextos,
2. *Benchmarking* em horários académicos,
3. Pesquisa local.

Nem toda a bibliografia pesquisada está diretamente relacionada com o desenvolvimento do trabalho em si, mas pretende dar um contributo para a compreensão e estruturação das ideias sobre o mesmo.

No subcapítulo 2.1 é abordado o problema dos horários académicos. Esta abordagem é iniciada através da explicação do conceito de *benchmarking* assim como a sua importância no desenvolvimento da dissertação. Mais concretamente através da análise de um documento sobre a formatação/definição de horários académicos, focando-se essencialmente nas diferentes restrições à geração de horários em diferentes países e também as utilizadas na especificação de horários na Academia da Força Aérea Portuguesa (AFA/FAP). Seguidamente no subcapítulo 2.2 é introduzido o formato XML para *benchmarking* referindo os tempos letivos, alunos, professores, salas, eventos e regras, sendo este demonstrado com um exemplo concreto. Finalmente no subcapítulo 2.3 são revistos os conceitos de pesquisa local restringida assim como algumas técnicas específicas, (por exemplo *Hill-Climbing*, *Simulated Annealing* e *Tabu Search*), sendo abordados alguns trabalhos sobre geração automática de horários que as utilizam.

2.1 Problema de Horários Académicos

Com alguma frequência se fala no meio académico e também empresarial no conceito de *benchmarking*. Este termo é definido em (da Silva, 2010) como a comparação de práticas e avaliação de desempenho de uma empresa em relação às outras. Visa também identificar padrões de boas práticas para aplicar nas avaliações e melhorar os desempenhos.

A Universidade de Twente (University of Twente, 2009a) tem vindo a desenvolver um projeto de *benchmarking* sobre horários académicos (University of Twente, 2009b),

combatendo a falta de estudos de *benchmarks* e formatos de dados sobre a temática dos horários académicos. A razão desta falta de informação advém da variedade de restrições e preferências dos ensinos, quer entre países diferentes ou até dentro do mesmo, quer entre diferentes sistemas de ensino, o que constitui um problema central na generalização de soluções para a elaboração de horários académicos.

A publicação referida em (Post, et al., 2012) foi desenvolvida no âmbito desse projeto e demonstra a necessidade de definição de um formato *standard* para *benchmarking* em horários académicos.

Este formato foi definido em *eXtensible Markup Language* (XML). O XML foi criado pela *World Wide Web Consortium* (W3C) e é um formato que permite organizar a informação hierarquicamente, delimitada por meta-informação e que pode ser entendida por utilizadores e por sistemas de informação (World Wide Web Consortium (W3C), 2013).

A geração de horários tem como principais orientações a organização da escola e a gestão de alunos, professores e salas. A natureza cultural, social e até mesmo a orientação política e económica de diferentes países influencia as regras e os objetivos na geração de horários. O projeto anteriormente mencionado, e que serve de guia de orientação desta dissertação, descreve essas mesmas regras e objetivos em diferentes países com a finalidade de alcançar requisitos comuns, e que tornem possível a criação do formato para *benchmarking* em horários académicos.

Para atingir os objetivos desta dissertação são descritos vários problemas na geração de horários por diferentes países, contemplando os seus requisitos e as diferenças entre eles, fornecendo assim informações credíveis para a compreensão e apoio na resolução do objetivo proposto. O enfoque do trabalho antes referido é relativo a escolas secundárias, mas as condicionantes gerais são semelhantes em instituições de ensino superior.

2.1.1 Austrália

Segundo (University of Twente, 2009c), as escolas públicas australianas têm dias letivos curtos, o que preenche a carga horária dos professores, evita a preocupação com a existência de tempos sem aulas, evita também a pré-atribuição de unidades curriculares a professores que só existirá para unidades curriculares específicas, obriga partilha de alguns cursos por dois professores e maximiza o uso de salas especiais, (por exemplo os laboratórios que possuem material específico).

As escolas privadas têm menos alunos e uma vasta escolha de unidades curriculares, tendo os professores mais tempo livre de aulas e as escolas mais salas especiais.

Objectivos e dificuldades na geração de horários académicos:

- Objectivo das escolas publicas é minimizar a partilha de alguns cursos por dois professores;
- Objectivo das escolas privadas é minimizar as aulas conjuntas com alunos de anos lectivos diferentes mas com o mesmo professor;
- Outra dificuldade tem a ver com a distribuição das aulas das unidades curriculares pelos cinco dias, devido a unidades curriculares escolhidas por alunos com número de aulas diferentes.

2.1.2 Reino Unido

A referência bibliográfica (University of Twente, 2009d) refere que se nos primeiros anos letivos das escolas do Reino Unido não há unidades curriculares optativas, (são todas obrigatórias), já nos restantes anos letivos são poucas as unidades curriculares obrigatórias.

As escolas dividem os alunos por grupos. Em algumas escolas, e devido à política escolar das mesmas, a cada grupo é atribuído um tutor, um professor com responsabilidades organizacionais e de encaminhamento. Em outras escolas é efetuado um agrupamento de outros grupos de alunos tutorados e as unidades curriculares são agrupadas em blocos de uma ou mais unidades curriculares de diferentes grupos de alunos e são organizadas num horário ao mesmo tempo.

Dificuldades na geração de horários académicos:

- Os alunos têm os horários sem tempos livres, exceto nos últimos anos escolares;
- Grande variedade de diferentes tempos de aulas;
- Um ou dois ciclos semanais de horários;
- Diferentes tempos letivos diários, e correspondente número de unidades curriculares, em diferentes escolas.

2.1.3 Finlândia

Em (University of Twente, 2009e) é referido que os horários académicos finlandeses são construídos para uma semana e são repetidos todo o ano letivo. As escolas finlandesas têm unidades curriculares obrigatórias e unidades curriculares optativas. Um aluno pertence a um

determinado grupo de alunos (turma) e pode pertencer a outros grupos adicionais apenas relacionados com unidades curriculares opcionais.

Dificuldades na geração de horários académicos:

- Os horários académicos finlandeses têm a particularidade de serem formados não por um conjunto de unidades curriculares, mas por um conjunto de aulas das unidades curriculares (lições);
- As lições são dadas a apenas um grupo de alunos, quer a unidade curricular seja obrigatória ou opcional;
- Só nos últimos anos letivos é que os professores são atribuídos a lições e a maioria das salas são reservadas consoante as decisões dos respetivos professores;
- Normalmente os anos letivos têm 10 unidades curriculares com uma carga horária entre duas a seis horas semanais, e as aulas podem durar entre uma e três horas;
- Os alunos têm os horários sem tempos livres.

2.1.4 Grécia

É descrito em (University of Twente, 2009f) que as aulas nas escolas gregas funcionam nos cinco dias da semana, têm entre 6 a 7 horas de aulas por dia, são pré-atribuídas aos professores, as aulas das unidades curriculares essenciais são distribuídas uniformemente pelos dias da semana e os alunos têm os horários sem tempos livres.

As dificuldades na geração de horários académicos derivam essencialmente das diferenças entre os primeiros e os últimos anos letivos, isto é:

Primeiros anos letivos

- Os grupos base de alunos mantêm-se por um determinado período do dia, para assim terem as aulas transversais aos cursos. Nos restantes momentos do dia os grupos de alunos são reorganizados de modo a frequentarem as aulas mais específicas, relativas às especializações escolhidas pelos alunos;
- É muito difícil organizar as unidades curriculares essenciais nas primeiras horas do dia;
- Também é muito difícil minimizar os tempos livres dos professores.

Últimos anos letivos

- Os grupos de alunos têm a maioria das aulas na mesma sala;

- Num período de tempo de aulas os grupos de alunos são divididos ou organizados com outros grupos de alunos para terem aulas de determinadas unidades curriculares. Nestas situações é normal haver dois professores, (ou em simultâneo ou para cada grupo de alunos divididos);
- As unidades curriculares essenciais devem ser nas primeiras horas do dia;
- Deve-se minimizar os tempos livres de aulas dos professores.

2.1.5 Holanda

Segundo (University of Twente, 2009g) nas escolas holandesas os professores são distribuídos por anos letivos. Nos primeiros anos os grupos de alunos têm as mesmas unidades curriculares, mas nos últimos os alunos têm unidades curriculares obrigatórias e opcionais consoante as especializações escolhidas.

Dificuldades na geração de horários académicos:

- Os horários são construídos para uma semana e são repetidos por seis semanas, um trimestre, semestre ou por todo o ano;
- Os alunos têm os horários sem tempos livres, exceto nos últimos anos escolares;
- Deve-se minimizar os tempos livres de aulas dos professores;
- Os professores são atribuídos a lições, por exemplo um professor para dar aulas teóricas e outro para aulas práticas.

O projeto referido em (University of Twente, 2009b) também teve o contributo dos problemas na geração de horários de países como Brasil, Dinamarca, Espanha, Itália, Kosovo e África do Sul, mas a grande diversidade dos problemas já está suficientemente ilustrada nos exemplos anteriores.

2.1.6 Horários na AFA/FAP

A Academia da Força Aérea, como academia de ensino superior, tem também requisitos para a geração de horários que dificultam a criação de uma metodologia para a sua geração automática. Numa academia militar de ensino superior em Portugal e focando essencialmente os alunos em formação interna na própria academia, são criados grupos de alunos consoante o ano letivo e a especialização.

Em função de informação informal obtida através do Gabinete de Estudos e Planeamento (GEP) da AFA foi possível identificar as principais dificuldades na geração de horários académicos, comparando-as com os exemplos anteriores:

- Os alunos têm os horários sem tempos livres entre aulas. Esta restrição é verificada em quase todos os primeiros anos letivos dos exemplos referidos anteriormente;
- Grande variedade de diferentes tempos de aulas, como acontece nas escolas do Reino Unido;
- Uma tarde por semana sem aulas, dedicada apenas a estudo e investigação, mais precisamente às quartas-feiras à tarde, ponto este que tem alguma semelhança com o ensino público australiano;
- Não há aulas às sextas-feiras à tarde. Como o anterior, este ponto tem alguma semelhança com o ensino publico australiano.
- As aulas de treino físico devem ser dadas nas últimas horas das manhãs ou tardes e devem ser distribuídas uniformemente pelos dias da semana, semelhante ao ensino na Grécia;
- As aulas de carácter militar devem ser dadas numa manhã específica da semana;
- Num período de tempo de aulas os grupos de alunos são divididos ou organizados com outros grupos de alunos para terem aulas de determinadas unidades curriculares. No caso de divisão, tem de haver mais que um professor para essas unidades curriculares, mas em caso de junção um professor dará a aula a mais que um grupo de alunos ao mesmo tempo;
- Os professores são atribuídos a lições.

Em termos gerais pode-se afirmar que o tipo de restrições entre os horários académicos da AFA é bastante semelhante ao das escolas gregas.

2.2 Formato para *benchmarking*

A análise da informação obtida sobre as várias dificuldades e requisitos dos horários académicos dos diferentes países permitiu a criação do formato em XML para *benchmarking* em horários académicos.

O formato XML contém a informação analisada anteriormente e essa informação está organizada hierarquicamente segundo conteúdos relativos a tempos letivos, alunos, professores, salas, eventos e regras.

2.2.1 Tempos letivos e recursos

Segundo (University of Twente, 2009b) os tempos letivos são aulas. Os grupos de tempos podem referir-se a tempos letivos por dia ou por semana e são frequentemente referidos nas validações das regras para a definição de um horário académico.

A criação de um horário académico útil não é possível só com tempos letivos, mas a associação de recursos a esses tempos já possibilita essa criação. Os recursos são as entidades aos quais são impostas regras. Neste contexto, são definidos como recursos os alunos, os professores e as salas de aulas.

- Os alunos ou grupos de alunos (ou até mesmo turmas) estão sujeitos as regras como ter ou não tempos livres entre aulas, número de aulas por dia e outras;
- Os professores podem estar afetos a unidades curriculares ou ser apenas especificado que as suas qualificações possibilitam lecionar aulas de algumas unidades curriculares. Esta informação é relevante para a implementação de regras como número de tempos livres, aulas por dia e até mesmo dias sem aulas;
- As salas de aulas têm especial relevância nos casos das salas especiais, por exemplo os laboratórios, ginásios, etc.

2.2.2 Eventos

A mesma referência bibliográfica especifica que os eventos são as aulas ou lições e atribuição de recursos a tempos letivos, tendo em conta as situações em que os professores não são pré-atribuídos às unidades curriculares. Neste caso após a escolha de um professor para a primeira aula de uma unidade curricular a um grupo de alunos, este deve manter-se nas restantes aulas da mesma unidade curricular e do mesmo grupo de alunos.

No formato XML, estão definidos eventos com as seguintes propriedades:

- A duração do evento;
- A unidade curricular relativa ao evento;
- O tempo letivo (hora de início) em que o evento está agendado;
- A carga horária do evento;
- O grupo de alunos do evento;
- O professor associado ao evento;
- Outros recursos, nomeadamente a sala, e até mesmo professores auxiliares entre outros.

2.2.3 Regras

As regras são definidas em (University of Twente, 2013h) como as condições existentes que podem ser utilizadas para restringir os horários académicos e no fim têm de estar satisfeitas no resultado obtido, isto é, no próprio horário.

As regras existentes são:

- *AssignResourceConstraint* – associa recursos a eventos;
- *AssignTimeConstraint* – associa tempos letivos a eventos;
- *SplitEventsConstraint* – limita o número de eventos de uma unidade curricular e as suas durações;
- *DistributeSplitEventsConstraint* – limita o número de eventos de uma certa duração de uma unidade curricular;
- *PreferResourcesConstraint* – especifica quais os recursos preferenciais de eventos;
- *PreferTimesConstraint* – especifica que alguns tempos letivos têm preferência na associação a eventos;
- *AvoidSplitAssignmentsConstraint* – obriga que os recursos estejam afetos a todos os eventos de uma unidade curricular, é o exemplo típico de um professor que dá todas as aulas de uma unidade curricular de um curso;
- *SpreadEventsConstraint* – especifica quais os eventos que devem ser distribuídos nos tempos letivos;
- *LinkEventsConstraint* – especifica que determinados eventos devem ser agendados ao mesmo tempo;
- *OrderEventsConstraint* – especifica que um evento só começa após um outro determinado evento terminar;
- *AvoidClashesConstraint* – esta é uma das regras com mais importância, evita conflitos de recursos;
- *AvoidUnavailableTimesConstraint* – especifica que alguns recursos estão indisponíveis para eventos a determinados tempos letivos;
- *LimitIdleTimesConstraint* – limita o número de tempos letivos que um recurso está livre;
- *ClusterBusyTimesConstraint* – limita o número de grupos de tempos letivos que um recurso está associado;

- *LimitBusyTimesConstraint* – limita o número de tempos letivos que um recurso está associado;
- *LimitWorkloadConstraint* – limita a carga horária total de certos recursos.

Todas as regras têm uma característica que influencia o cálculo do custo da violação da mesma, nomeadamente o desvio (*deviation*). Nem todas as regras calculam o desvio segundo a mesma premissa. Normalmente é considerado como desvio a duração da aula.

Existem propriedades comuns a todas as regras, que têm como objetivo identificar inequivocamente a regra e catalogar o peso da mesma para permitir que seja calculado o custo inerente.

As propriedades comuns existentes são:

- *Id* – serve para identificar inequivocamente a regra, permitindo assim que a mesma seja referenciada;
- *Name* – serve para identificar a regra em modo visual, isto é, serve para os utilizadores poderem identificar a regra num modo legível;
- *Required* – propriedade booleana que determina se a regra é uma restrição obrigatória ou não;
- *Weight* – peso dado à regra, inteiro de 1 a 1000 inclusive;
- *CostFunction* – função para o custo da regra que é aplicada ao respetivo valor do desvio. Há 3 funções possíveis para a propriedade *CostFunction*:
 - *Linear* – o valor da função é o valor do desvio;
 - *Quadratic* – o valor da função é o quadrado do desvio;
 - *Step* – o valor da função é 0 ou 1 se o desvio for diferente de 0;
- *AppliesTo* – limita os eventos e/ou recursos sobre os quais a regra é aplicada.

Caso a regra seja definida como uma restrição obrigatória, isto é *Required* = True, é atribuído ao respetivo custo um valor muito elevado, caso contrário o custo da regra é calculado através da fórmula:

$$Cost = Weight * CostFunction(deviation)$$

Segundo (University of Twente, 2013h), a existência de apenas estas regras não é limitativa, sendo possível criar mais regras de modo a contribuir para necessidades diferentes em gerações de horários. Esta informação torna-se importante devido a possíveis necessidades diferentes em gerações de horários académicos para ensino superior. A existência de

departamentos académicos que reúnem unidades curriculares assim como professores pode dar origem a novas regras.

Estão exemplificadas algumas regras (restrições) no A.2.

2.2.4 Formato XML

A Universidade de Twente divulgou em (University of Twente, 2013h) o formato XML, que tal como referido anteriormente este formato contem toda a informação necessária para a interpretação de horários académicos, mais precisamente informação relativa a tempos letivos, alunos, professores, salas, eventos e regras.

A informação está organizada segundo a hierarquia da informação, e para criar tal organização foi definida uma estrutura da qual o formato tem de obedecer.

O formato está definido do seguinte modo:

- *Archives* – coleção de instâncias de horários académicos, mais concretamente um *archive* é o aglutinador de todos os horários académicos definidos num mesmo ficheiro XML;
- *Instances* – é uma ocorrência de um horário académico num determinado tempo escolar, isto é, um horário académico dum ano/semestre/trimestre. As *instances* estão contidas num *archive*;
- *Times* – definem os dias de semana aplicáveis aos horários, as semanas que incluem os respetivos dias, e outros tipos de tempos que podem, por exemplo, conter *TimeGroups* para uma melhor organização. Um caso concreto é a definição de manhãs ou tardes como *TimeGroups*. Os *Times* estão contidos nas *Instances*;
- *Resources* – declaram as entidades aos quais são impostas regras, mais concretamente os alunos, os professores e as salas de aulas, e estão contidos nas *Instances*;
- *Resources Groups* – são coleções de recursos, (dentro de *Resources*), que referenciem tipos de recursos com regras específicas;
- *Events* – como referido anteriormente são as aulas ou lições às quais é necessário atribuir *Resources* em determinados *Times* e também estão contidos nas *Instances*;
- *Constraints* – são as regras definidas anteriormente, isto é, as restrições impostas às *Instances*, e como tal estão contidas nelas;

- *Solution Groups* – são coleções de soluções, isto é, são concretamente o aglutinador de todas as soluções geradas para os horários académicos definidas num mesmo ficheiro XML, as *Solution Groups* estão contidas num *archive*;
- *Solutions* – são as soluções para um respetivo horário, as *Solutions* estão contidas no *Solution Groups*;
- *Reports* – é toda a informação obtida após a geração do horário, estão contidas em *Solution* e entre outras informações retêm o custo da geração do horário assim como as regras violadas.

De seguida é apresentado um breve extrato XML retirado do apêndice A.1:

```
<HighSchoolTimetableArchive>
  <Instances>
    <Instance Id="3">
      <Times>
        <TimeGroups>
          <Day Id="2">
            <Name>Segunda</Name>
            <TimeGroup Id="Mornings">
              <Name>Manhãs</Name>
            </TimeGroup>
          </Day>
        </TimeGroups>
        <Resources>
          <ResourceTypes>
            <ResourceType Id="Teacher">
              <ResourceGroups>
                <ResourceGroup Id=" AllTeachers">
                  <Resource Id="1T0016866">
                    <Name>MANUEL M. DE</Name>
                    <ResourceType Reference="Teacher"/>
                    <ResourceGroups>
                      <ResourceGroup Reference="AllTeachers"/>
                    </ResourceGroups>
                  </Resource>
                </ResourceGroup>
              </ResourceGroups>
            </ResourceType>
          </ResourceTypes>
          <Events>
            <EventGroups>
              <Course Id="11101">
                <Name>Mestrado em Aeronáutica Militar, Especialidade de Piloto Aviador</Name>
              </Course>
            </EventGroups>
            <Event Id="19">
              <Name>OTOPCM</Name>
              <Duration>1</Duration>
              <Course Reference="11101"/>
            </Event>
          </Events>
        </Resources>
      </Instance>
    </Instances>
  </HighSchoolTimetableArchive>
```

```

<Resources>
  <Resource Reference="PILAV_1">
    <Role>Class</Role>
    <ResourceType Reference="Class"/>
</SolutionGroups>
<SolutionGroup Id="AFA_2014-01-24">
  <Solution Reference="3">
    <Events>
      <Event Reference="TEACH_1_PILAV_1_1">
        <Duration>1</Duration>
        <Time Reference="21" />
      </Event>
      <Time Reference="Monday_1" />
    </Events>
  </Solution>
</SolutionGroup>
</SolutionGroups>

```

A informação existente no formato XML é passível de ser usada para geração de novos horários académicos, mais concretamente através de técnicas de pesquisa local.

2.3 Pesquisa Local

Em (Gaspero, 2003), a pesquisa local é referida como pertencente à família de técnicas usadas em problemas de pesquisa e otimização (Brailsford, Potts, & Smith, 1999), isto é, dado um conjunto de variáveis, um conjunto finito de valores possíveis para essas variáveis e uma lista de restrições, descobrir valores para as variáveis que satisfaçam as restrições.

A pesquisa local é um método para encontrar soluções para problemas de otimização sujeita a restrições, que se baseia na melhoria iterativa dos valores atribuídos às variáveis até todas as regras (restrições) estarem salvaguardadas (ou pelo menos as regras obrigatórias) e a função objetivo esteja (quase) otimizada, com o custo zero ou pelo menos reduzido. A pesquisa local pode assim ser considerada como um método de pesquisa que começa num estado inicial, obtido através de uma escolha de dados aleatória (ou heurísticamente por semelhança com casos anteriores) e que em cada iteração é melhorado através de pequenas modificações (Rossi, Beek, & Walsh, 2006).

Para este trabalho é, portanto, necessário clarificar o processo de pesquisa local restringida como um processo que tem como objetivo alcançar uma solução que satisfaça as restrições impostas às variáveis. Em (Brailsford, Potts, & Smith, 1999) está exemplificado o caso da geração de horários.

```

function Local Search (Search Space  $S$ , Neighborhood  $N$ , Cost Function  $F$ )
begin
     $s_0 := InitialSolution(S)$ ;
     $i := 0$ ;
    while ( $\neg StopCriterion(si, i)$ ) do
        begin
             $m := SelectMove(si, F, N)$ ;
            if ( $AcceptableMove(m, si, F)$ )
                then swap( $m, si$ );
             $i := i + 1$ 
        end
    end return  $si$ ;

```

Ilustração 1: Algoritmo de Pesquisa Local

A Ilustração 1, baseada no algoritmo de pesquisa local existente em (Gaspero, 2003), apresenta um algoritmo abstrato de pesquisa local. Nesta função estão referidos os parâmetros de entrada “*Search Space*”, “*Neighborhood*” e “*Cost Function*”.

- *Search Space* – refere o espaço de estados que podem ser gerados ou melhorados durante a pesquisa. A solução final, (no caso dos horários, os tempos letivos atribuídos às aulas), retornada pela função é o estado final obtido na pesquisa;
- *Neighborhood* (Vizinhança) – é o conjunto de estados para o qual um determinado estado pode transitar, (normalmente por alteração do valor de uma ou mais variáveis). No caso dos horários pode representar alterações aos recursos atribuídos às aulas (alunos, professores, salas) e os tempos letivos em que as aulas são lecionadas;
- *Cost Function* – é a função que calcula os custos de cada estado, baseado nas restrições violadas e no objetivo a atingir.

O algoritmo tende a melhorar ao longo do processamento a qualidade das soluções exploradas, obtidas por avaliação do grau de satisfação das restrições e no final do processamento devolve o último estado obtido.

O processamento abstrato de pesquisa local da Ilustração 1 é iniciado através da função “*InitialSolution*” que escolhe aleatoriamente ou heurísticamente o primeiro estado a ser manipulado. Uma vez escolhido esse estado é começado o ciclo que só terminará quando se

satisfaz o critério de paragem. A escolha dos valores que irão alterar as variáveis é baseada nos custos das alterações induzidas nos estados obtidos.

As funções abstratas utilizadas no processamento descrito anteriormente são materializadas dependendo das técnicas e algoritmos utilizados.

Várias são as técnicas que implementam a pesquisa local e que têm como objetivo atingir uma solução que salvaguarda as regras impostas mas com o menor custo possível.

Há dois tipos extremos de técnicas para alcançar este objetivo durante o ciclo iterativo. No primeiro, as alterações dos valores das variáveis tentam sempre diminuir ou pelo menos não aumentar o custo dos estados seguintes (técnicas *Greedy*). O segundo, utiliza métodos aleatórios para alteração das variáveis (técnicas *Random*), (Rossi, Beek, & Walsh, 2006). A razão destes dois tipos prende-se com uma lacuna das técnicas *Greedy*, que não conseguem obter melhores vizinhos quando o estado atingido é um mínimo local ou uma superfície plana (*Plateau*). As técnicas *Random* permitem ultrapassar este problema efetuando movimentos aleatórios.

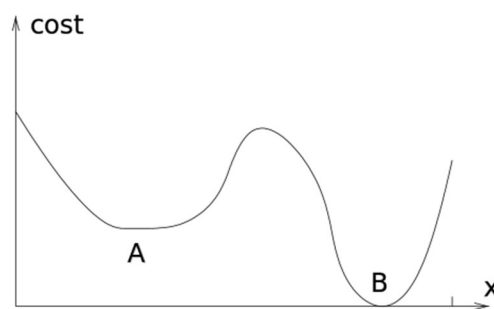


Ilustração 2: Exemplo de *Plateaus* em Pesquisa Local

A Ilustração 2 exemplifica um problema de minimização em que no espaço de estados existe um *Plateau* que inclui o estado A e em que o processo não consegue prosseguir até ao estado B apenas com técnicas *Greedy*.

Os algoritmos de pesquisa local utilizam geralmente heurísticas a curto prazo para melhorar (ou não piorar) o estado corrente, (técnicas *Greedy*), e mais raramente aceitam piores estados, (técnicas *Random*). Tipicamente são utilizadas meta-heurísticas que permitem aceitar estados piores para melhorar a longo prazo as soluções obtidas.

Das várias técnicas que implementam a pesquisa local serão referidas algumas variantes de *Hill-Climbing* (HC) e as técnicas *Simulated Annealing* (SA) e *Tabu Search* (TS). Outras técnicas como os algoritmos genéticos e as colónias de formigas não serão abordadas nesta

dissertação por se tratarem de técnicas baseadas em populações e que utilizam métodos bastantes distintos.

2.3.1 *Hill-Climbing*

Hill-Climbing é uma das técnicas mais simples de pesquisa local que tenta otimizar a função objetivo, (em geral, nesta dissertação a técnica *Hill-Climbing* foi adaptada para minimizar a função objetivo). Como a generalidade de algoritmos de pesquisa local, este algoritmo não mantém em memória uma árvore de pesquisa, guardando apenas o estado (ou nó) corrente, (Russell & Norvig, 1995).

O *Hill-Climbing* é uma técnica que devolve em cada iteração um estado com melhor valor (menor custo) na vizinhança do estado corrente.

Durante a pesquisa local com a técnica de *Hill-Climbing* há duas situações que devem ser referidas:

- *Local minima* (Mínimo local) – um mínimo local é um estado com custo mais baixo que os seus vizinhos, mas possivelmente mais alto que outros estados, nomeadamente a um mínimo global. Sempre que o algoritmo atinja um tal estado ele já não atingirá o mínimo global, mesmo que o algoritmo ainda não tenha atingido um resultado satisfatório;
- *Plateaux* (*Planaltos*) – tal como referido anteriormente, esta situação acontece quando os vizinhos do estado corrente têm um valor semelhante.

Caso alguma destas situações seja atingida, então a melhor solução adaptada é geralmente reiniciar o processamento a partir de outro estado inicial. *Random-restart hill-climbing* (RRHC) é uma variante de *Hill-Climbing* que adota exatamente esta solução, podendo-se definir o número de iterações a utilizar. *Random-restart hill-climbing* implementa uma meta-heurística aleatória.

Utilizando como referência o exemplo demonstrado na Ilustração 1, na implementação de *Random-restart hill-climbing*, a função *AcceptableMove* aceita um estado cujo custo seja preferencialmente menor que o estado anterior, ou então um estado com custo não pior que o estado anterior. Para o processamento atingir uma solução final, passando em certos casos por mínimos locais ou planaltos, as funções *InitialSolution* e *StopCriterion* são manipuladas, de

modo a solução inicial não ser sempre a mesma e no caso dos planaltos o critério de paragem utilizar uma constante com o número máximo de iterações.

É conhecida a utilização de outras variantes do *Hill-Climbing* em trabalhos sobre geração automática de horários, mais concretamente as variantes *Late Acceptance Hill-Climbing* (LAHC) e *Step Counting Hill-Climbing* (SCHC). Estas variantes previnem os mínimos locais e os planaltos.

2.3.1.1 *Late Acceptance Hill-Climbing*

Esta variante de *Hill-Climbing* é explicada em (Bykov & Burke, A Late Acceptance Strategy in Hill-Climbing for Exam Timetabling Problems, 2008) como tendo a particularidade de implementar uma heurística quase *Greedy*, com a diferença de aceitar um vizinho melhor que um dos estados (ou nós) visitados em K ciclos anteriores.

Na mesma referência bibliográfica, os criadores de *Late Acceptance Hill-Climbing*, (Yuri Bykov e Edmund K. Burke), demonstram a eficiência do algoritmo aplicando-o na geração automática de horários para exames e comparam os resultados obtidos entre esta técnica e o tradicional *Hill-Climbing*.

Para implementar *Late Acceptance Hill-Climbing* no exemplo presente na Ilustração 1, a função *AcceptableMove* aceita um estado cujo custo fosse menor que o estado anterior ou menor que estados tratados em determinados ciclos anteriores.

2.3.1.2 *Step Counting Hill-Climbing*

A variante *Steps Counting Hill-Climbing* é introduzida em (Bykov & Petrovic, A Step Counting Hill Climbing algorithm, 2013) como uma evolução de *Late Acceptance Hill-Climbing*. Esta evolução surge depois de constatado o sucesso da variante anterior e a tentativa de a tornar mais eficiente mas mantendo as suas vantagens. A diferença está na não existência de uma lista com algumas soluções anteriores, mas a continuação da necessidade de aceitar um vizinho melhor, ou pelo menos melhor que um estado (ou nó) anterior, considerado com o valor limite (“*cost bound*”).

Yuri Bykov, que foi coautor de *Late Acceptance Hill-Climbing*, e Sanja Petrovic são os criadores de *Steps Counting Hill-Climbing* e demonstram em (Bykov & Petrovic, A Step

Counting Hill Climbing algorithm, 2013) a eficiência do algoritmo aplicando-o na geração automática de horários para exames e comparam os resultados obtidos entre esta técnica, *Late Acceptance Hill-Climbing* e *Simulated Annealing*.

Tendo novamente como base o exemplo demonstrado na Ilustração 1, a implementação de *Steps Counting Hill-Climbing* obriga a função *AcceptableMove* a aceitar um estado cujo custo seja preferencialmente menor que o estado anterior ou pelo menos menor que um determinado estado anterior (estado com o valor limite).

2.3.2 *Simulated Annealing*

Simulated Annealing é uma técnica de pesquisa local probabilística sendo o seu nome justificado por analogia com a termodinâmica (Gaspero, 2003). Em (Russell & Norvig, 1995) esta técnica é referida como um meio alternativo ao reinício de um processamento devido a um mínimo local. Basicamente em vez de escolher um estado cujo custo é menor ou igual ao estado anterior, esta técnica gera e avalia movimentos aleatórios.

No processamento da técnica *Simulated Annealing* é usado um parâmetro que representa a temperatura (T). T começa com um valor elevado que vai sendo diminuído ao longo das iterações. Esta variável é manipulada por uma função que determina a velocidade à qual ela é reduzida em consideração ao número de ciclos ocorridos.

A implementação desta técnica no exemplo demonstrado na Ilustração 1 impõe que as funções *InitialSolution* e *SelectMove* façam sempre uma escolha aleatória, que a função *StopCriterion* seja verdade quando T atinge o valor 0, (ou um valor suficientemente baixo), e a função *AcceptableMove* aceite incondicionalmente estados cujo custo seja inferior ao anterior, ou, no caso contrário, essa aceitação é condicional (com probabilidade $e^{-\Delta E/T}$, em que ΔE representa a alteração dos custos do estado corrente e vizinho).

A heurística *Random* é basicamente aplicada em *Simulated Annealing* por gerar um qualquer vizinho, sendo a meta-heurística aplicada para aceitar probabilisticamente (com menor probabilidade no final do processo pois T desce ao longo da pesquisa) piores valores para o estado vizinho que é gerado.

O trabalho explicado em (Zhang, Liu, M'Hallah, & Leung, 2009) utiliza a técnica *Simulated Annealing* para geração automática de horários e tem como particularidade a execução do algoritmo em duas fases. Na primeira fase é gerada uma possível solução inicial

e na segunda fase é alcançada uma solução boa. A segunda fase é alcançada utilizando trocas de tempos letivos. Este trabalho refere que este algoritmo teve uma performance melhor que o algoritmo tradicional (com uma só fase).

Em (Ceschia, Gaspero, & Schaerf, 2012) e (Bellio, Ceschia, Gaspero, Schaerf, & Urli, 2013) estão referidos dois trabalhos que também utilizam a técnica *Simulated Annealing*. A diferença destes dois trabalhos está relacionada com a origem das restrições e o objetivo dos mesmos, isto é, o trabalho referido em (Ceschia, Gaspero, & Schaerf, 2012) está relacionado com a aplicação de restrições sobre os alunos e respetivos cursos, enquanto no trabalho referido em (Bellio, Ceschia, Gaspero, Schaerf, & Urli, 2013) todas as restrições e objetivos estão relacionados com a configuração inicial de um curso sem a aplicação de restrições sobre alunos.

2.3.3 *Tabu Search*

A técnica *Tabu Search*, como referida em (Gaspero, 2003), tem a particularidade de manter informação sobre soluções visitadas mais recentemente. Para implementar esta ideia é utilizada uma lista Tabu, lista essa referida em (Thamilselvan & Balasubramanie, 2009) como uma memória de curto prazo que contém informação das soluções que foram visitadas (menos de k iterações atrás onde k , o número de soluções a armazenar, é chamado de *tabu tenure*).

A técnica *Tabu Search* pode ser implementada como o exemplo presente na Ilustração 1. A função *SelectMove* seleciona estados vizinhos e a função *AcceptableMove* aceita o estado obtido se este não pertencer à lista Tabu. Este estado é armazenado na lista Tabu e caso a lista esteja cheia o estado com mais tempo na lista Tabu é retirado.

Em certas exceções é utilizada uma função de aspiração que permite selecionar estados presentes na lista Tabu. Esta situação só é possível se o custo seja inferior ao custo do estado corrente.

O *Tabu Search* implementa a heurística ao selecionar bons vizinhos e implementa meta-heurística ao impedir que alguns bons vizinhos Tabu sejam aceites e explorados repetidamente. Mais concretamente, seleciona bons vizinhos através de trocas com a vizinhança, mas considera apenas os bons vizinhos que não estejam referidos na lista Tabu.

Este algoritmo também é referido em trabalhos sobre geração automática de horários, nomeadamente os trabalhos referidos em (Dorneles, 2010) e (Subramanian, Medeiros,

Formiga, & Souza, 2011) com objetivo semelhante ao desta dissertação. Estes trabalhos têm a particularidade de a solução inicial ser gerada através de algoritmos que não pretendem alcançar soluções ótimas. O trabalho referido em (Dorneles, 2010) tem também como premissa a informação guardada num ficheiro com um formato padrão em XML, o que realça ainda mais a aproximação com esta dissertação.

2.3.4 Outros

Para além das técnicas já referidas existem também trabalhos sobre geração automática de horários que utilizam outras técnicas tais como os algoritmos genéticos em (Ni & Yang, 2013) e (Yang & Jat, 2011) e colónias de formigas em (Thepphakorn, Hicks, & Pongcharoen, 2014) e (Nothegger, Mayer, Chwatal, & Raidl, 2012).

3 TimeTabling

TimeTabling foi o nome dado ao sistema por exemplificar o verdadeiro objetivo do mesmo, gerar horários académicos. É um sistema acedido através de interface *Web*, utiliza um servidor aplicacional (Barry & Associates, Inc., 2000) como fornecedor dos seus serviços e foi desenvolvido com uma metodologia que possibilita o uso do seu código no desenvolvimento de sistemas acedidos de outras maneiras. Este sistema tem também como valências, a possibilidade de criar ficheiros com formato em XML para *benchmarking* em horários académicos a partir de informação existente em repositórios de dados, fazer upload de ficheiros que obedecem às mesmas regras e posterior seleção dos mesmos, visualizar detalhadamente a informação e obter indicadores da performance das técnicas utilizadas nas gerações dos horários.

No contexto da dissertação foi necessário considerar linguagens que fossem eficientes na resolução de problemas com restrições e que implementassem pesquisa local. COMET (Dynadec, 2010) é uma linguagem especializada nestas resoluções que foi considerada, mas para além de ser uma linguagem proprietária não é uma linguagem utilizada pela FAP.

O sistema foi desenvolvido na linguagem JAVA (Oracle, 2007). É uma linguagem de código aberto, orientada a objetos, é executada através de uma máquina virtual o que a torna portátil, sendo a sua sintaxe similar a C/C++. Esta linguagem é utilizada pela FAP no desenvolvimento de meros aplicativos mas também de sistemas de informação complexos. A FAP é dotada de sistemas próprios em JAVA que funcionam de forma autónoma mas também de sistemas acedidos através de interface WEB. A interface WEB foi desenvolvida em JSP (Oracle, 2013) e javascript (W3Schools, 1999).

No seu funcionamento, o trabalho tem um ficheiro com formato em XML para *benchmarking* em horários académicos a funcionar como repositório de dados e um sistema com interface WEB que implementa as técnicas de pesquisa local abordadas, (*Simulated Annealing*, *Tabu Search* e as variantes de *Hill-Climbing*), para gerar automaticamente horários académicos.

Os algoritmos referentes às técnicas antes referidas têm como solução inicial um horário previamente concebido e presente no ficheiro XML ou então geram uma solução inicial. Em

caso de inexistência de uma solução é sempre gerada uma solução inicial, onde são gerados horários preenchidos com tempos letivos e recursos seleccionados aleatoriamente. As classes que interpretam as regras/restrições do problema devolvem o custo do estado vizinho aos respetivos algoritmos.

As variáveis tratadas são referentes aos tempos letivos, unidades curriculares, salas, professores e turmas. Estas variáveis estão todas definidas no ficheiro XML e são utilizadas pelo sistema WEB.

Relativamente ao universo de dados tratados, foram manipuladas 5 turmas, cada turma com cerca de 35 aulas por semana, (7 horas vezes 5 dias), que envolveu a atribuição de estados com mais de 500 variáveis.

O desenvolvimento seguiu o modelo *Model-view-controller* (MVC) (Cunningham & Cunningham, Inc., 2013). Modelo este que isola o ambiente gráfico do controlador das regras do sistema e da base de dados, o que juntamente com uma linguagem de programação adequada, possibilita a criação de sistemas modernos eficientes para interfaces WEB, aplicativos locais e até aplicações para dispositivos móveis, este modelo está ilustrado na Ilustração 3.

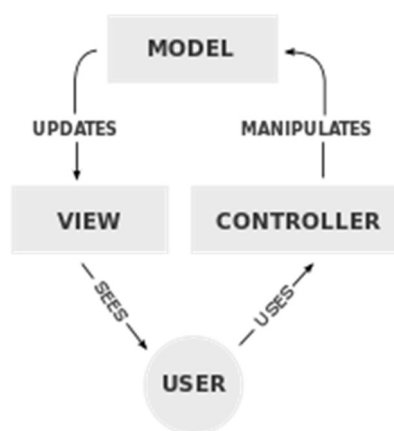


Ilustração 3: Modelo MVC

3.1 Implementação MVC

A implementação MVC no sistema é facilmente identificada ao aceder ao código através de um editor e está demonstrada na Ilustração 4.

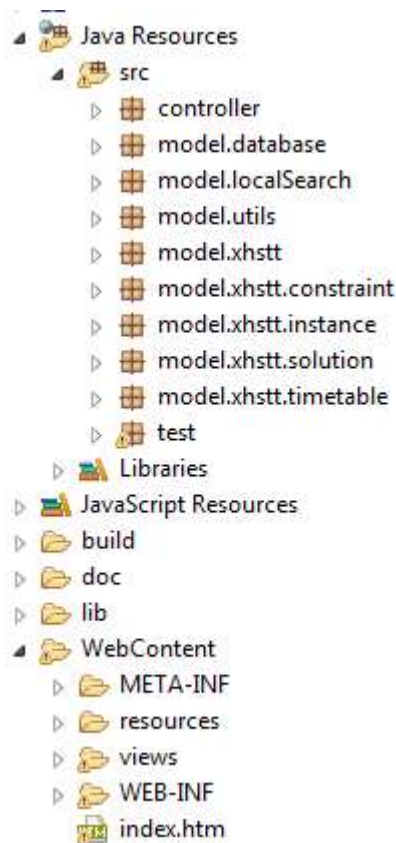


Ilustração 4: Código do TimeTabling

3.1.1 *Controller*

Controller, está definido no sistema como um pacote que engloba as classes responsáveis por fazer a ligação entre a camada de visualização e a lógica do sistema. Estas classes funcionam como *servlet*, processando requisições (*request*), respostas (*response*) e gerando ou invocando páginas HTML ou JSP. As classes criadas são as seguintes:

- `AuthenticationServlet.java`: Permite gerir a autenticação de utilizadores;
- `CreateFileServlet.java`: Classe responsável por receber parâmetros com a descrição de consultas a base de dados e assim criar ficheiros com formato em XML para *benchmarking* em horários académicos;
- `CreateSolutionServlet.java`: Esta é a classe invocada quando é solicitada a geração de horários académicos;
- `SelectFileServlet.java`: Permite selecionar ficheiros armazenados no sistema;
- `UploadDownloadFileServlet.java`: Permite armazenar ficheiros;
- `WB_GetSolutionEventsConstraintServlet.java`: Devolve as restrições violadas numa solução;

- WB_GetSolutionEventsServlet.java: Devolve os eventos da solução;
- WB_GetSolutionGroupServlet.java: Devolve grupos de soluções;
- WB_GetTimeTablesServlet.java: Devolve os horários.

3.1.2 *Model*

Model, ou mais concretamente, a lógica do sistema, é onde estão definidas todas as classes responsáveis pela estratégia utilizada para entre outras coisas ser possível gerar horários. As classes *Model* estão organizadas estruturalmente de modo a permitir uma gestão mais prudente, uma codificação mais eficiente e uma ligação entre as classes também muito eficiente.

Organização das classes:

- database - pacote com o intuito de armazenar a classe responsável por aceder a base de dados:
 - JDBCParser.java: classe onde utiliza uma estrutura enumerada (*Enum*) para implementar configurações estandardizadas para aceder a base de dados;
- localSearch - pacote crucial para a geração automática de horários académicos:
 - LocalSearch.java: classe onde as técnicas de pesquisa local estão desenvolvidas, são acedidas através de uma função básica de pesquisa local utilizando mecanismos de rescrita e organizadas segundo uma estrutura enumerada;
 - LocalSearchInterface.java: *interface* Java onde estão definidas os métodos reescritos pela classe LocalSearch.java;
 - Neighborhood.java: classe responsável por fornecer a vizinhança de cada estado;
- utils - pacote que guarda as classes utilitárias do sistema:
 - Authentication.java: classe responsável pela informação necessária para a autenticação dos utilizadores;
 - Pair.java: classe que permite criar pares de objetos;
 - XmlDomDocument.java: classe utilizada para interagir com ficheiros XML;
- xhstt - pacote onde se encontram as classes que interpretam o formato em XML para *benchmarking* em horários académicos.

- XHSTT.java: classe que permite gerir a informação existente no ficheiro com o formato em XML para *benchmarking*;
- XHSTT_DB.java: classe que entende a classe XHSTT.java, possibilitando a criação do ficheiro com informação existente numa base de dados;
- constraint - pacote com as classes responsáveis pela implementação das restrições impostas aos horários:
 - Constraint.java: classe que implementa a estrutura base das restrições, e os métodos generalistas relativos às restrições;
 - ConstraintInterface.java: interface com as definições dos métodos de “Deviation” reescritos pela classe ConstraintTypes.java;
 - ConstraintTypes.java: classe onde estão caracterizados os vários tipos de restrições, utiliza métodos de reescrita e devolvem o custo inerente a cada restrição violada para cada aula instanciada;
 - CostFunctionInterface.java: interface com as definições dos métodos de “CostFunction” reescritos pela classe CostFunctionTypes.java;
 - CostFunctionTypes.java: classe que através de métodos de reescrita implementa as funções de custo associadas a cada restrição definida no ficheiro;
 - EventPair.java: classe que permite gerir informação referente a certos tipos de restrições;
- instance - pacote onde estão definidas todas as classes que instanciam os objetos existentes nas instâncias definidas no ficheiro:
 - Course.java: curso;
 - Event.java: unidade curricular;
 - Event_EventGroup.java: grupos de eventos a que pertence o evento;
 - Event_Resource.java: recursos afetos ao evento;
 - Event_ResourceGroup.java: grupos de recursos afetos ao evento;
 - EventGroup.java: grupo de eventos;
 - Instance.java: instância;
 - Resource.java: recurso;
 - Resource_ResourceGroup.java: grupos de recursos a que pertence o recurso;
 - ResourceGroup.java: grupo de recursos;

- ResourceType.java: tipo de recursos;
- Time.java: tempo letivo;
- Time_TimeGroup.java: grupos de tempos letivos a que pertence o tempo letivo;
- TimeGroup.java: grupos de tempos letivos;
- solution - pacote onde estão definidas todas as classes que instanciam os objetos existentes nas soluções definidas no ficheiro:
 - Solution.java: solução;
 - SolutionEvent.java: aula;
 - SolutionEvent_Constraint.java: restrição da aula;
 - SolutionEvent_Resource.java: recurso da aula;
 - SolutionGroup.java: grupo de soluções;
 - SolutionReport.java: relatório da solução;
 - SolutionReport_Statistic.java: estatística da solução;
- timetable - pacote com as classes pertinentes para a demonstração de horários:
 - TimeTable.java: classe que implementa os métodos para interagir com um horário;
 - TimeTableSlot.java: aula existente num horário.

Estas classes podem ser utilizadas em outros sistemas que implementem o método MVC, mesmo que os sistemas utilizem outro tipo de interface.

3.1.3 *View*

View, é a componente do método MVC onde estão definidas as propriedades visuais do sistema, materializadas na interface gráfica.

A camada visual foi desenvolvida em JSP com a inclusão de jQuery e do template Bootstrap.



Ilustração 5: Entrada no sistema

A Ilustração 5 demonstra o acesso ao sistema, onde é possível identificar o objetivo do sistema e inserir as credenciais que permitem aceder às funcionalidades do mesmo.

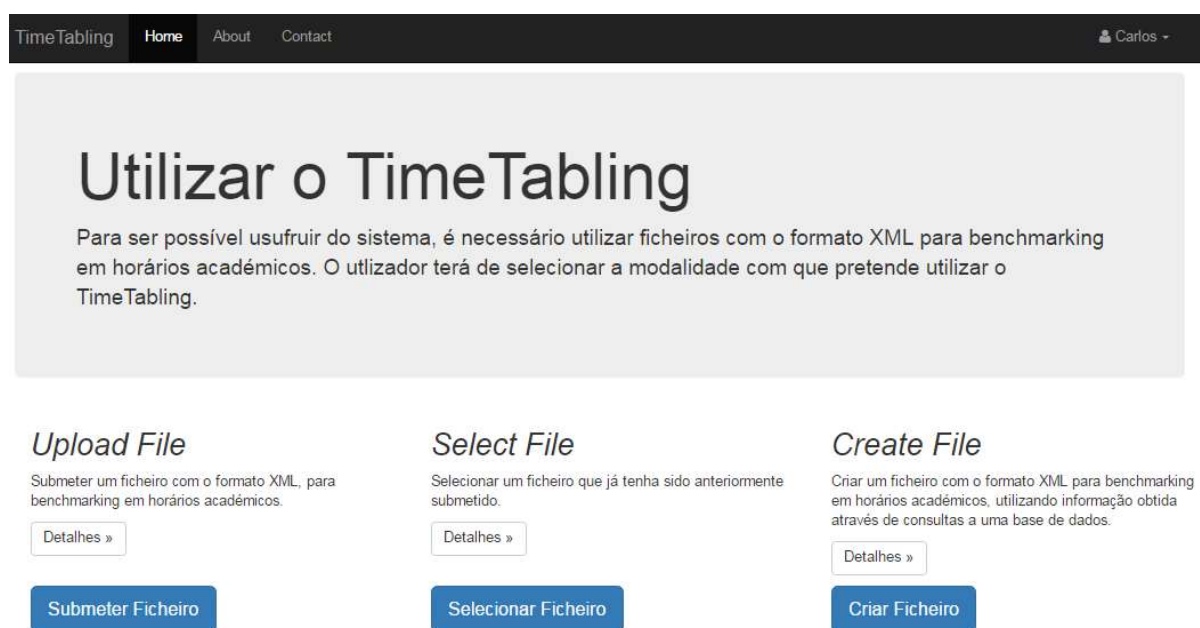


Ilustração 6: Ecrã inicial do sistema

É no ecrã apresentado na Ilustração 6 que o utilizador seleciona as funcionalidades que permitem aceder ao detalhe de uma instância escolar:



Ilustração 7: Upload de ficheiros



Ilustração 8: Seleção de ficheiros

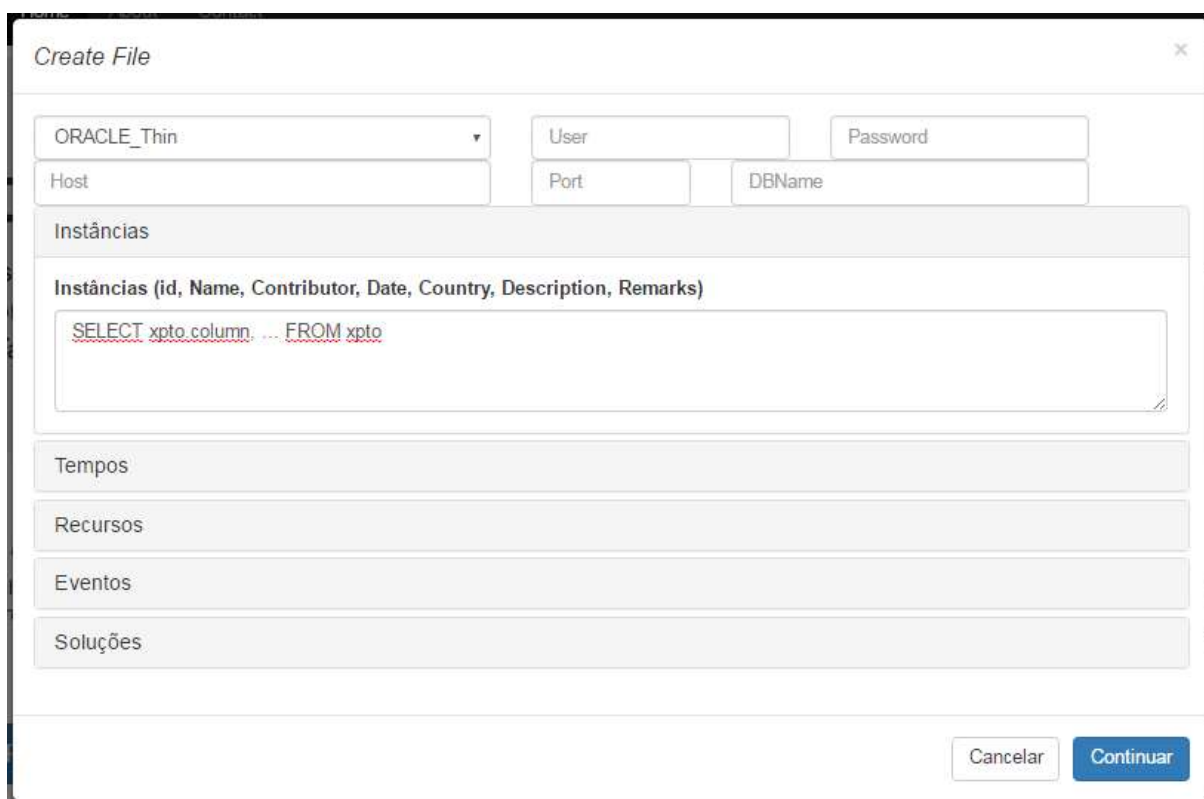


Ilustração 9: Criação de ficheiros

Depois de selecionar qualquer uma das três funcionalidades antes referidas, o utilizador acede ao detalhe da instância. O detalhe da instância é a informação da origem da instância assim como a informação dos cursos, eventos, recursos e restrições da mesma.

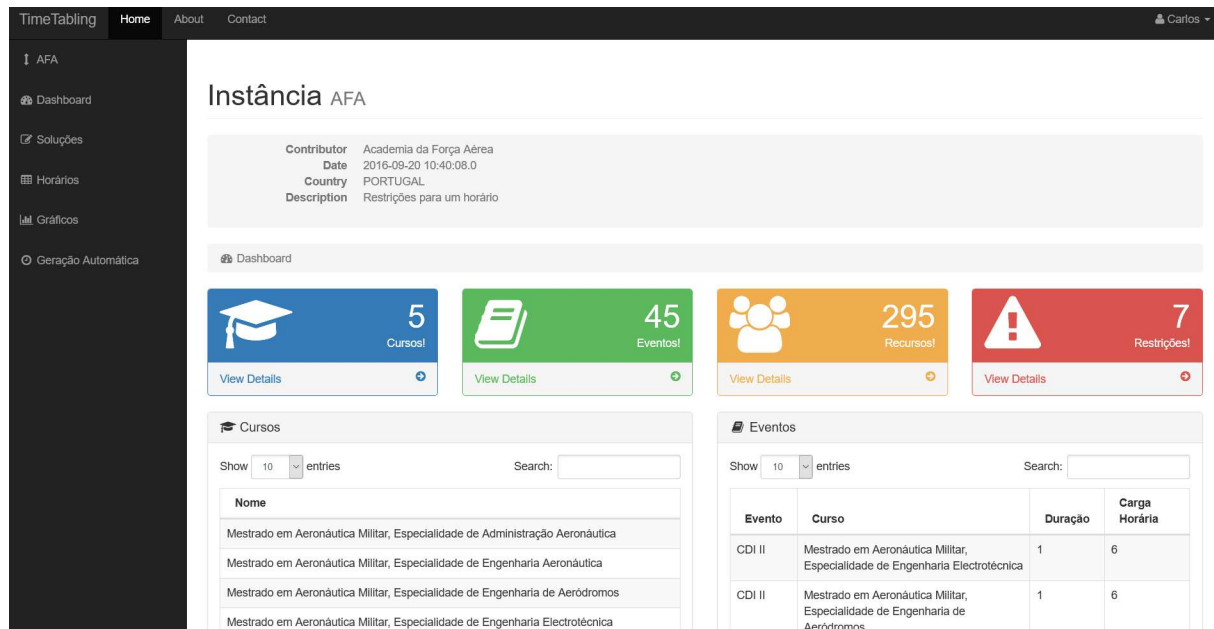


Ilustração 10: Detalhe da instância

Na Ilustração 10 está evidenciado o menu lateral que permite um acesso rápido a determinada informação:

- *Dashboard*: informação dos cursos, eventos, recursos e restrições definidos na instância;
- *Soluções*: detalhe das soluções existentes no ficheiro assim como as possíveis violações das restrições, tem a possibilidade de navegar entre soluções;
- *Horários*: horários por solução;
- *Gráficos*: gráficos por número de restrições, estados verificados, segundos e custos;
- *Geração automática de horários académicos*: através das técnicas programadas.

Na mesma ilustração é possível verificar que no detalhe da instância estão visíveis os cursos e os eventos dessa mesma instância.

Continuando o detalhe da instância também estão visíveis os recursos e as restrições, como está exemplificado na Ilustração 11.

Recursos	
Show 10 entries	Search: <input type="text"/>
Nome	Tipo
ADMAER	Class
AFONSO A. E	Teacher
ALBERTO B. GAMA	Teacher
ALBERTO C. SILVA	Teacher
ALBERTO L. BENTO	Teacher
ALBERTO L. SANTOS	Teacher
ALBERTO M. MARQUES	Teacher
ALBERTO N. AMOROSO	Teacher
ALBERTO N. RIBEIRO	Teacher
ALBERTO R. MARTINS	Teacher
Showing 1 to 10 of 295 entries	Previous 1 2 3 4 5 ... 30 Next

Restrições		
Show 10 entries	Search: <input type="text"/>	
Nome	Obrigatório	Peso
AssignTimes	true	1
No_Idle_Times_For_Classes	false	1
NoAvailableTimesFrid	true	1
NoAvailableTimesWedAfter	true	1
NoAvailableTimesWeek	true	1
NoResourceClashes	true	1
PreferredTimesEFI	false	1
Showing 1 to 7 of 7 entries	Previous 1 Next	

Ilustração 11: Continuação do detalhe da instância

Para cada instância pode haver várias soluções, na Ilustração 12 está demonstrado o detalhe de uma solução.

Soluções	
Contributo	Carlos de Passos
Data	20/09/2016 03:04:11
Descrição	Geração sem grandes restrições
Publicação	
Observações	
Algoritmo	Random-restart hill-climbing
Tempo de execução	01:41:493
← « 1 2 » →	

Eventos na Solução						
Show 10 entries	Search: <input type="text"/>					
Curso	Evento	Data	Hora	Duração	Custo	
Mestrado em Aeronáutica Militar, Especialidade de Administração Aeronáutica	HEE	Quarta	11:10	1	0	
Mestrado em Aeronáutica Militar, Especialidade de Administração Aeronáutica	HEE	Quarta	08:00	1	0	
Mestrado em Aeronáutica Militar, Especialidade de Administração Aeronáutica	HFF	Segunda	16:40	1	0	

Ilustração 12: Detalhe da solução

O detalhe da solução é a informação pertinente da solução em que entre outras coisas refere o algoritmo/técnica que a gerou e o tempo de execução. Ainda no seguimento do

detalhe da solução é possível visualizar todos os eventos da mesma assim como as restrições violadas.

Nas Ilustração 13 está demonstrado um horário de um curso, dividido por dias da semana e tempos letivos. Em cada aula está visível qual a unidade curricular, o professor e a sala.

O horario visível é o produto final da geração automática usando um algoritmo de pesquisa local com restrições às aulas.

Mestrado em Aeronáutica Militar, Especialidade de Piloto Aviador					
	Segunda	Terça	Quarta	Quinta	Sexta
08:00	IGE MANUEL G. AZEVEDO SALA14	OPA FILIPE A. DE SALA12	CDI II PAULA J. SILVA LCB25	IGE MANUEL G. AZEVEDO SALA14	CDI II PAULA J. SILVA LCB25
09:00	IGE MANUEL G. AZEVEDO SALA14	OPA FILIPE A. DE SALA12	CDI II PAULA J. SILVA LCB25	IGE MANUEL G. AZEVEDO SALA14	CDI II PAULA J. SILVA LCB25
10:10	CDI II PAULA J. SILVA LCB25	MTC GODINHO S. DA SALA4	MTC GODINHO S. DA SALA4	EDM ALBERTO N. AMOROSO LCB23	EFI I MANUEL M. MOTA PAV_DESP
11:10	CDI II PAULA J. SILVA LCB25	MTC GODINHO S. DA SALA4	MTC GODINHO S. DA SALA4	EDM ALBERTO N. AMOROSO LCB23	
13:30	LIG II MANUEL R. DA LCB23	FMA I FILIPE F. SERAFIM LCB40		LIG II MANUEL R. DA LCB23	FMA I FILIPE F. SERAFIM LCB40
14:30	OPA FILIPE A. DE SALA12	FMA I FILIPE F. SERAFIM LCB40		LIG II MANUEL R. DA LCB23	
15:40	OTOPCM MIGUEL M. CABEÇA SALA1	EFI I MANUEL M. MOTA PAV_DESP			
16:40				EFI I MANUEL M. MOTA PAV_DESP	

Ilustração 13: Horário

Após a geração de um horário é possível ver gráficos que informam sobre os detalhes na execução da técnica que o gerou, assim como das gerações anteriores. Os gráficos existentes são referentes ao número de restrições violadas por estados verificados e por segundos e aos custos por número de restrições violadas e por segundos. Em cada gráfico existe uma legenda no canto superior direito com a abreviatura da técnica responsável pela geração do horário.

A Ilustração 14 é um exemplo de um gráfico com o número de restrições por segundos de uma solução gerada pela técnica *Random-restart hill-climbing* e outra pela técnica *Tabu Search*.

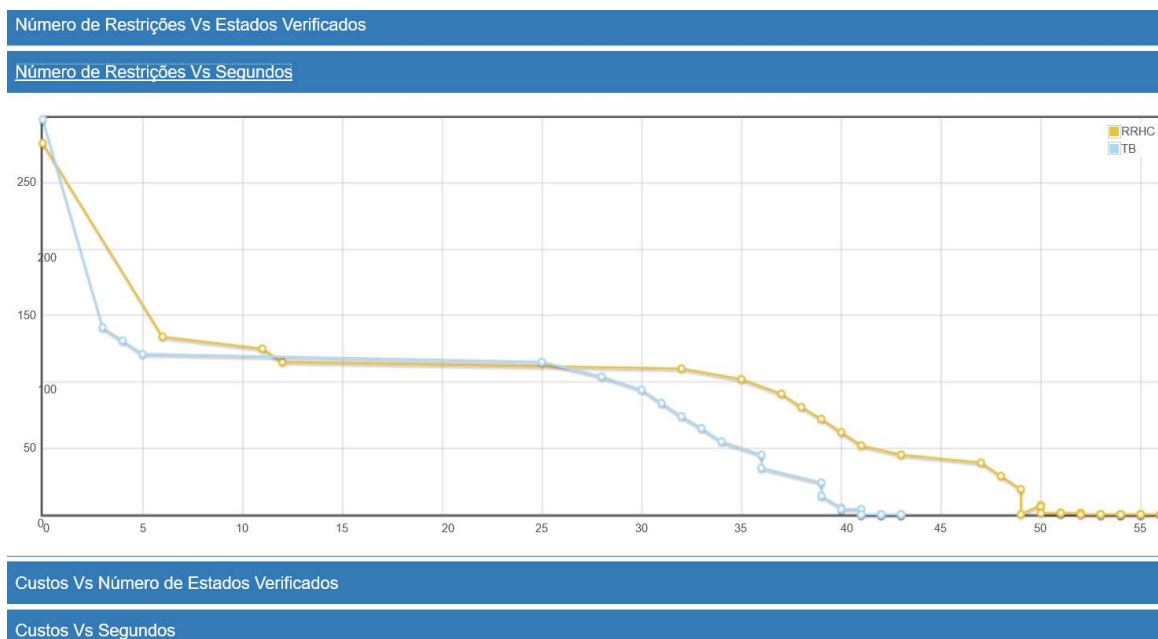


Ilustração 14: Gráficos

A Ilustração 15 exemplifica o pedido da geração automática. O utilizador seleciona a técnica responsável pela geração e depois edita o detalhe da solução. Nesta edição está disponível a opção de ser gerada uma nova solução aleatória ou uma geração baseada numa solução existente.

Geração automática

Técnica: Simulated Annealing

Contributo:

Descrição:

Publicação:

Observações:

Baseado na Solução:

RRHC LAHC SCHC SA TB

Ilustração 15: Geração automática

3.2 Plano do trabalho

O plano de trabalho para a elaboração da dissertação seguiu a seguinte sequência de tarefas:

1. Criação de classes que instanciam os objetos existentes no ficheiro;
2. Criação das classes que implementam as restrições;
3. Criação das classes que implementam a pesquisa local;
4. Criação da interface web;
5. Implementação das estatísticas.
6. Análise dos resultados obtidos.
7. Conclusão da escrita do relatório da dissertação.

3.3 Implementação da Pesquisa Local

Foram implementadas as cinco técnicas de pesquisa local abordadas. A codificação está presente na “LocalSearch.java”, e é através desta classe que a solicitação para a geração de horários assim como a própria geração está desenvolvida.

A função exemplificada na Ilustração 1 está presente na classe “LocalSearch.java”, a sua implementação é básica e é a função base das técnicas de pesquisa local. Os métodos “InitialSolution”, “StopCriterion”, “SelectMove” e “AcceptableMove” são reescritos consoante a técnica de geração desejada.

Para uma melhor performance, foram utilizadas no desenvolvimento do sistema, estruturas existentes na linguagem de programação JAVA, que melhoram a pesquisa e o armazenamento da informação. Mais concretamente estrutura *hash*, *set*, *list*, *array*, etc...

3.3.1 *Random-restart hill-climbing*

- InitialSolution: este método é muito idêntico em todas as técnicas, tem como particularidade ser uma solução antes gerada ou então uma solução gerada aleatoriamente, devido às características desta técnica, foi introduzida a

inicialização de variáveis que auxiliam a sua execução, mais concretamente sobre o a possibilidade de reiniciar a função;

- **StopCriterion:** o critério de paragem da técnica é atingir um número máximo de iterações ou atingir a solução ideal, a solução com custo zero, a solução sem violação de restrições;
- **SelectMove:** o estado desejado é alcançado através de procura na vizinhança de um estado não pior que o estado atual, esta procura só é executada caso o custo do estado atual seja maior que 0, esta procura passa primeiro por validar a troca do estado tendo em consideração apenas os tempos letivos sem aulas atribuídas, (o que beneficia o alcance de uma solução sem furos entre aulas), caso não encontre um estado aceitável considera a troca de tempos letivos entre o estado atual e os estados na vizinhança e caso ainda seja necessário poderá validar a troca de recursos, (salas, professores, etc...);
- **AcceptableMove:** são aceites as trocas que não piorem o custo, mas caso a procura de estados melhores seja continuamente não alcançada, então aceita um estado com custo pior.

3.3.2 *Late Acceptance Hill-Climbing*

- **InitialSolution:** uma solução antes gerada ou então uma solução gerada aleatoriamente;
- **StopCriterion:** o critério de paragem da técnica é atingir um número máximo de iterações ou atingir uma solução ideal;
- **SelectMove:** o estado desejado é alcançado do mesmo modo que a técnica *Random-restart hill-climbing* mas com a particularidade de procurar na vizinhança de um estado não pior que o estado atual ou que K estados visitados anteriormente;
- **AcceptableMove:** são aceites as trocas que não piorem o custo.

3.3.3 *Step Counting Hill-Climbing*

- InitialSolution: uma solução antes gerada ou então uma solução gerada aleatoriamente;
- StopCriterion: o critério de paragem da técnica é atingir um número máximo de iterações ou atingir uma solução ideal;
- SelectMove: o estado desejado é alcançado do mesmo modo que a técnica *Random-restart hill-climbing* mas com a particularidade de procurar na vizinhança de um estado não pior que o estado atual ou que um determinado estado visitado anteriormente;
- AcceptableMove: são aceites as trocas que não piorem o custo.

3.3.4 *Simulated Annealing*

- InitialSolution: uma solução antes gerada ou então uma solução gerada aleatoriamente, devido às características desta técnica, foi introduzida a inicialização de variáveis que auxiliam a sua execução, mais concretamente sobre a variável T, (temperatura);
- StopCriterion: o critério de paragem da técnica é atingir uma temperatura muito reduzida;
- SelectMove: o estado desejado é alcançado através de troca de variáveis existentes na vizinhança;
- AcceptableMove: são aceites as trocas que não piorem o custo ou então que o custo não seja superior à função:

$$\text{Math.exp}((\text{NewState.getCost()} - \text{ActualState.getCost()}) / T);$$

O método “AcceptableMove” termina com a função responsável por fazes decrescer a temperatura:

$$T *= 1 - \text{CoolingRate};$$

3.3.5 *Tabu Search*

- InitialSolution: uma solução antes gerada ou então uma solução gerada aleatoriamente;
- StopCriterion: o critério de paragem da técnica é atingir um número máximo de iterações ou atingir uma solução ideal, a solução com custo zero, a solução sem violação de restrições;
- SelectMove: o estado desejado é alcançado através de troca de variáveis existentes na vizinhança que não estejam presentes na lista Tabu, na lista Tabu estão referenciados os índices dos estados visitados em trocas anteriores;
- AcceptableMove: as trocas são aceites e a lista Tabu é atualizada com o índice do estado melhorado.

3.4 Material utilizado

O TimeTabling foi desenvolvido na linguagem de programação JAVA, com o JDK na versão 8.0_92.

Foi utilizado JSP e javascript para a programação da interface, com jQuery na versão v1.11.3 e a inclusão da biblioteca bootstrap na versão v3.3.6. Foram também utilizados os plugins de jQuery: bootpag na versão 1.0.7 no componente que permite a navegação entre soluções e javascript plotting library na versão 0.8.3 para a implementação dos gráficos.

As ferramentas utilizadas foram os editores Eclipse (Eclipse Foundation, s.d.) e Notepad++ (Notepad++, s.d.) assim como o servidor aplicacional (Barry & Associates, Inc., 2000) Wildfly (Red Hat, Inc., 2013). O editor Eclipse na versão Mars.2 Release (4.5.2) para a programação do sistema, assim como para configuração do servidor aplicacional utilizado, Wildfly na versão 9.0.2.

O editor Notepad++ na versão v6.9.2, foi utilizado para em determinadas situações auxiliar a compreensão/edição de ficheiros.

4 Resultados Experimentais

Foram efetuados cinco tipos de experiências, em cada tipo de experiência foram executadas as cinco técnicas de pesquisa local.

Devido à particularidade de cada técnica, foram usados parâmetros específicos para as referidas técnicas, como está demonstrado na Tabela 1.

Nome	Valor	Descrição	Técnica que usa
iRequiredCost	1000000	Custo das validações obrigatórias	Todas
iMaxCicle	Nº de aulas	Nº máximo de ciclos	<i>Random-restart hill-climbing;</i> <i>Late Acceptance Hill-Climbing;</i> <i>Step Counting Hill-Climbing</i>
iMaxSameIterator	2	Nº máximo de ciclos sem alterações de custo total	<i>Random-restart hill-climbing</i>
iKStates	2	Nº de estados tratados anteriormente ao estado corrente, que validam o custo do vizinho	<i>Late Acceptance Hill-Climbing</i>
iCostBoundIndex	i	Variável com o índice do outro estado a validar o custo	<i>Step Counting Hill-Climbing</i>
iCostBoundStates	10	Nº de iterações até alterar a variável iCostBoundIndex	<i>Step Counting Hill-Climbing</i>
iInitialTemp	1000	Temperatura inicial	<i>Simulated Annealing</i>
dCoolingRate	0.003	Nº que influência a frequência em que a temperatura baixa	<i>Simulated Annealing</i>
iTemperature	$iTemperature * 1 - dCoolingRate$	Variável com a temperatura, inicia com a variável iInitialTemp	<i>Simulated Annealing</i>
iTabuSize	10	Tamanho da lista tabu	<i>Tabu Search</i>

Tabela 1: Parâmetros utilizados das técnicas de pesquisa local

Os resultados experimentais só são possíveis através de uma evolução ao formato *benchmark*, mais concretamente com a criação da estrutura “*Statistics*”. Nesta estrutura é possível guardar em qualquer momento específico da geração da solução, o número de restrições violadas, o custo da solução e o tempo de execução.

4.1 Baseadas em horários existentes

Foram gerados horários baseados no horário do 1º semestre do ano letivo de 2011/2012 da AFA, onde foram utilizadas 5 turmas, cada turma com cerca de 35 aulas por semana, (7 horas vezes 5 dias).

No caso dos horários da AFA foram consideradas para as restrições base os seguintes tipo de regras:

- *AssignTimeConstraint*: Para obrigar a que cada aula tenha um tempo letivo atribuído;
- *AvoidClashesConstraint*: Para evitar conflitos de recursos;
- *AvoidUnavailableTimesConstraint*: Para evitar que sejam atribuídas aulas em determinados tempos letivos;
- *PreferTimesConstraint*: Para indicar que determinadas unidades curriculares sejam lecionadas em determinados tempos letivos;
- *LimitIdleTimesConstraint*: Para limitar o número de tempos letivos livres entre aulas.

Os testes experimentais ocorreram com todas as técnicas implementadas. Para cada técnica o sistema executou cinco vezes como está exemplificado na Ilustração 16.

RRHC	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	5621	00:05,621	-1422,4	00:01,422	0	0
2	30000000	30	5721	00:05,721	-1322,4	00:01,322	0	0
3	30000000	30	8081	00:08,081	1037,6	00:01,038	0	0
4	30000000	30	7904	00:07,904	860,6	00:00,861	0	0
5	30000000	30	7890	00:07,890	846,6	00:00,847	0	0
	Média		7043,4	00:07,043			0	0
	Desvio Padrão		1255,58	00:01,256			0,00	0,00

Ilustração 16: Resultados experimentais com a técnica *Random-restart hill-climbing*

4.1.1 Alteração da tarde livre para outro dia

Foi alterada a restrição que impossibilitava haver aulas às quartas-feiras, trocando o dia de quarta-feira para terça-feira, deste modo o sistema teve de alocar as aulas existentes na terça-feira noutros tempos letivos, validando a ocorrência de violações de restrições.

Os resultados experimentais estão presentes no apêndice A.3. A média dos resultados experimentais por técnica estão ilustrados na Ilustração 17.

Técnicas	Tempo de Execução		Valores Finais	
	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	7043,4	00:07,043	0	0
LAHC	8172,6	00:08,173	0	0
SCHC	8242,4	00:08,242	0	0
SA	101351,2	01:41,351	3200002	5,2
TS	8176,8	00:08,177	0	0

Ilustração 17: Média dos resultados experimentais da alteração da tarde livre

Neste teste as variantes de *Hill-climbing* e *Tabu Search* demonstraram melhores resultados.

Teoricamente o que havia a fazer neste caso era mudar as aulas de terça-feira para quarta-feira. Na prática é um pouco idêntico ao que as variantes de *Hill-climbing* e *Tabu Search* fazem, quando descobrem tempos livres para colocar as aulas.

A performance da técnica *Random-restart hill-climbing* foi a melhor, muito provavelmente pela codificação simples de apenas avaliar o custo do estado vizinho num horário com poucas restrições definidas.

A técnica *Simulated Annealing* foi a única que apresentou resultados finais com custos elevados e restrições violadas. Demonstrou que não é a ideal para estes exemplos, pois o facto de validar apenas estados seleccionados aleatoriamente, prejudica a rápida seleção de estados que teoricamente se sabe que têm menor custo.

Embora esteja evidenciada a performance da técnica *Random-restart hill-climbing*, é conveniente alertar que pela própria natureza, as variantes de *Hill-climbing* podem em determinadas situações entrar em *loop*, mais concretamente quando as trocas de estados são a três, isto é, o estado na posição A troca com o estado na posição B que por sua vez troca com C e este volta a trocar com o A e assim sucessivamente. Deste modo é conveniente referir que pela sua natureza, a técnica *Tabu Search* dá garantias de fiabilidade.

4.1.2 Indisponibilidade de dois professores para um dia

Foi criada a restrição que impede dois professores de darem aulas num determinado dia. Os resultados experimentais estão presentes no apêndice A.4. A média dos resultados experimentais por técnica estão ilustrados na Ilustração 18.

Técnicas	Tempo de Execução		Valores Finais	
	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	9186	00:09,186	0	0
LAHC	9312	00:09,312	0	0
SCHC	9069,2	00:09,069	0	0
SA	27549,4	00:27,549	200000,8	1
TS	8877	00:08,877	0	0

Ilustração 18: Média dos resultados experimentais com a indisponibilidade de dois professores

Também neste teste a técnica *Simulated Annealing* demonstrou que não é a ideal. Mais uma vez apresentou resultados finais com custos elevados e restrições violadas, mas com melhores resultados que o teste referido anteriormente. Para este teste há a necessidade de avaliar se após a troca entre estados não há choques entre recursos.

As variantes de *Hill-climbing* e *Tabu Search* demonstraram novamente melhores resultados. Embora esteja salientada a performance da técnica *Tabu Search*. Esta situação ocorre devido à não seleção de estados visitados recentemente, o que previne as trocas as três e por sua vez tenta melhorar os custos dos estados com mais celeridade.

4.1.3 Indisponibilidade de uma sala toda a semana

Foi criada a restrição que coloca uma sala indisponível toda a semana. Os resultados experimentais estão presentes no apêndice A.5. A média dos resultados experimentais por técnica estão ilustrados na Ilustração 19.

Técnicas	Tempo de Execução		Valores Finais	
	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	71176,8	01:11,177	0	0
LAHC	89793,2	01:29,793	0	0
SCHC	77289,4	01:17,289	0	0
SA	111148,6	01:51,149	9000000,4	9,4
TS	74600,6	01:14,601	0	0

Ilustração 19: Média dos resultados experimentais com a indisponibilidade de uma sala

Neste teste as performances pioram em geral. Este é o tipo de caso que há a necessidade de substituir o recurso que está indisponível, o que se manifesta em mais operações e portanto piora a performance.

Mais uma vez o facto de validar apenas estados seleccionados aleatoriamente, faz com que a técnica *Simulated Annealing* tenha novamente uma má performance.

A técnica *Tabu Search*, evidencia novamente a melhor performance, pelas mesmas razões que foram referidas anteriormente.

4.2 Através de geração aleatória de horários

Os testes experimentais ocorreram com todas as técnicas implementadas. Foram gerados horários aleatórios para a AFA e para o *benchmark* de uma escola da Grécia.

Os testes experimentais ocorreram com todas as técnicas implementadas.

4.2.1 Horários para a AFA

Para cada técnica foram gerados dez horários aleatórios.

Os resultados experimentais estão presentes no apêndice A.6. A média dos resultados experimentais por técnica estão ilustrados na Ilustração 20.

Técnicas	Tempo de Execução		Valores Finais	
	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	72224,6	01:12,225	0	0
LAHC	61921,5	01:01,922	0	0
SCHC	483287,1	08:03,287	200000,1	0,3
SA	156660,8	02:36,661	2200003,3	4,7
TS	55245,8	00:55,246	0	0

Ilustração 20: Média dos resultados experimentais da geração aleatória de horários para a AFA

Foi possível apurar neste teste que mais uma vez a técnica *Tabu Search* evidencia a melhor performance e que a técnica *Simulated Annealing* tem a pior performance.

As variantes de *Hill-climbing* demonstraram bons resultados em geral, embora a técnica *Step Counting Hill-Climbing* tenha obtido restrições violadas em dois testes, como está demonstrado no apêndice A.6, esta situação deverá estar relacionada com o caso das trocas de estados a três.

O grande número de restrições violadas no início da geração, faz com que o processamento seja mais demorado que os testes referidos anteriormente.

4.2.2 Horários para benchmark de escola da Grécia

Foi utilizado o *benchmark* (Tassopoulos & Beligiannis, 2011) sobre uma instância de uma Universidade na Grécia com 184 aulas. Para este teste o foram gerados horários em todas as técnicas implementadas. Para cada técnica o sistema executou cinco vezes.

Neste caso foram consideradas para as os seguintes tipo de regras:

- *AssignTimeConstraint*: Para obrigar a que cada aula tenha um tempo letivo atribuído;
- *AvoidClashesConstraint*: Para evitar conflitos de recursos;
- *AvoidUnavailableTimesConstraint*: Para evitar que sejam atribuídas aulas determinados tempos letivos;
- *SpreadEventsConstraint*: Para indicar quais as unidades curriculares que devem ser distribuídos nos tempos letivos;
- *LimitIdleTimesConstraint*: Para limitar o número de tempos letivos livres entre aulas;
- *LimitBusyTimesConstraint*: Para limitar o número de tempos letivos que um recurso está associado.

Os resultados experimentais estão presentes no apêndice A.7. A média dos resultados experimentais por técnica estão ilustrados na Ilustração 21.

Técnicas	Tempo de Execução		Valores Finais	
	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	207184,6	03:27,185	0	0
LAHC	191478,2	03:11,478	0	0
SCHC	203581,6	03:23,582	0	0
SA	753510	12:33,510	30600000	30,6
TS	186964,6	03:06,965	0	0

Ilustração 21: Média dos resultados experimentais da geração aleatória de horários para a Grécia

A técnica *Tabu Search* está novamente em evidência, com a melhor performance, a técnica *Simulated Annealing* tem novamente a pior performance.

As variantes de *Hill-climbing* demonstraram bons resultados em geral, com resultados muito próximos da técnica *Tabu Search*.

O grande número de restrições violadas no início da geração somando ao grande número de restrições existentes no *benchmark*, faz com que o processamento seja demorado.

Face à inexistência de informação sobre o tempo de execução da solução inicial existente no atual *benchmark*, não é possível comparar os tempos de execução entre a solução gerada pelo *Timetabling* e pelo sistema que gerou a solução inicial existente. Deste modo, é apenas possível garantir que a solução inicial existente não tinha restrições violadas e que o *Timetabling* também consegue obter soluções finais sem restrições violadas.

5 Conclusões

A geração manual de horários requer trabalho árduo devido à necessidade de gerir os diversos conflitos de restrições impostos aos recursos a ser usados.

O *Timetabling* é um sistema que foi desenvolvido para colmatar a dificuldade da geração manual de horários. Foi desenvolvido na linguagem JAVA, para ser acedido através de interface Web.

O sistema permite para além da geração automática de horários, (aleatória ou baseada num horário existente), a visualização detalhada dos recursos, restrições, horários, gráficos de performance da geração dos horários entre outras coisas.

Para funcionar corretamente o *Timetabling* necessita da informação estruturada num ficheiro XML com o formato para *benchmarking* de horários académicos.

A geração automática de horários é realizada através de técnicas de pesquisa local restringida, mais concretamente as variantes de *Hill-climbing*, (*Random-restart hill-climbing*, *Late Acceptance Hill-Climbing* e *Step Counting Hill-Climbing*), a técnica *Simulated Annealing* e a técnica *Tabu Search*.

Após vários testes é possível concluir que o sistema reúne todas as condições para funcionar na FAP, mais concretamente na AFA, mas que pode ser utilizado noutras instituições das Forças Armadas com outro tipo de restrições.

É também possível concluir que a técnica *Tabu Search* dá garantias de fiabilidade e eficiência e teve em geral a melhor performance nos testes realizados. Em contrapartida a técnica *Simulated Annealing*, nos moldes em que teoricamente está definida e por conseguinte codificada no sistema, demonstrou em termos gerais que tem a performance mais fraca.

De futuro, pretende-se que o sistema seja verificado em mais instituições, militares ou não, de modo a garantir a fiabilidade do sistema.

A interface do sistema pode evoluir, mais concretamente através de funcionalidades de criação de restrições, apoiar a geração de horários validando instâncias trivialmente impossíveis através de verificações da viabilidade dos recursos existentes, (verificando por exemplo se o número total de salas é inferior às necessárias), indicadores para salvaguardar

margem de segurança, (ex. 20% de horas de salas disponíveis a mais que o necessário), entre outras.

O *Timetabling* pretende ser uma ajuda na geração automática de horários através de técnicas de pesquisa local restringida e pretende contribuir para o uso de um formato em XML para *benchmarking* de horários académicos. Embora o sistema concretize o que estava estipulado neste trabalho, ele ainda tem margem para evoluir. Esta evolução dará robustez e a garantia de um sistema fiável e eficiente.

Bibliografia

- Bellio, R., Ceschia, S., Gaspero, L. D., Schaerf, A., & Urli, T. (2013). A simulated annealing approach to the curriculum-based course timetabling problem. Ghent: MISTA 2013.
- Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 19, 557–581.
- Bykov, Y., & Burke, E. K. (2008). A Late Acceptance Strategy in Hill-Climbing for Exam Timetabling Problems. Montréal: Practice and Theory of Automated Timetabling.
- Bykov, Y., & Petrovic, S. (2013). A Step Counting Hill Climbing algorithm. *Nottingham University Business School*.
- Ceschia, S., Gaspero, L. D., & Schaerf, A. (2012). Design, Engineering, and Experimental Analysis of a Simulated Annealing Approach to the Post-Enrolment Course Timetabling Problem. *Computers & Operations Research*, 39(7), 1615-1624.
- da Silva, S. T. (2010). Direito Administrativo Europeu. Em S. T. da Silva, *Direito Administrativo Europeu* (p. 88). Coimbra: Imprensa da Universidade de Coimbra.
- Dorneles, Á. (2010). *Desenvolvimento de um Aplicativo para Geração Automática de Quadros de Horários Escolares*. Santa Maria: Universidade Federal de Santa Maria.
- Dynadec. (2010). *Comet Tutorial*. Providence: Dynamic Decision Technologies Inc.
- Gaspero, L. D. (2003). *Local Search Techniques for Scheduling Problems: Algorithms and Software Tools*. Udinese: Forum Edizioni.
- Ni, J., & Yang, N.-N. (2013). Genetic Algorithm and Its Application in Scheduling System. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(4), 1934-1939.
- Nothegger, C., Mayer, A., Chwatal, A., & Raidl, G. R. (2012). Solving the post enrolment course timetabling problem by ant colony optimization. *Annals of Operations Research*, 149(1), 325-339.

- Post, G., Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngas, J., Nurmi, C., & Ranson, D. (2012). An XML format for benchmarks in High School Timetabling. *Annals of Operations Research*, 194, 385-397.
- Rossi, F., Beek, P. V., & Walsh, T. (2006). Handbook of Constraint Programming. Em F. Rossi, P. V. Beek, & T. Walsh, *Handbook of Constraint Programming* (p. 245). Amsterdam: Elsevier.
- Russell, S. J., & Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. New Jersey: Prentice Hall.
- Subramanian, A., Medeiros, J. M., Formiga, L. d., & Souza, M. J. (2011). Aplicação da Metaheurística Busca Tabu ao Problema de Alocação de Aulas a Salas em uma Instituição Universitária. *Revista Produção Online*, 11(1), pp. 54-75.
- Thamilselvan, R., & Balasubramanie, P. (2009). Integrating Genetic Algorithm, Tabu Search Approach for Job Shop Scheduling. *International Journal of Computer Science and Information Security*, 2, 134-139.
- Thepphakorn, T., Hicks, C., & Pongcharoen, P. (2014). An ant colony based timetabling tool. *International Journal of Production Economics*, 109, 131-144.
- Yang, S., & Jat, S. N. (2011). Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1), 93-106.
- Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. C. (2009). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203, 550-558.

Webliografia

- Barry & Associates, Inc. (2000). *Application Server Definition*. Obtido em 10 de 01 de 2016, de Service Architecture: http://www.service-architecture.com/articles/application-servers/application_server_definition.html
- Cunningham & Cunningham, Inc. (30 de 09 de 2013). *Model View Controller*. Obtido em 16 de 12 de 2014, de WikiWikiWeb: <http://c2.com/cgi/wiki?ModelViewController>
- Eclipse Foundation. (s.d.). *Eclipse*. Obtido em 10 de 01 de 2016, de Eclipse: <https://eclipse.org/>
- Notepad++. (s.d.). *Notepad++ Home*. Obtido em 10 de 01 de 2016, de Notepad++: <https://notepad-plus-plus.org/>
- Oracle. (2007). *Java*. Obtido em 30 de 12 de 2013, de Java: <https://www.java.com>
- Oracle. (31 de 12 de 2013). *JavaServer Pages Technology*. Obtido em 10 de 01 de 2016, de Java: <https://www.oracle.com/technetwork/java/javaee/jsp/index.html>
- Red Hat, Inc. (2013). *WildFly Homepage*. Obtido em 10 de 01 de 2016, de WildFly: <http://wildfly.org/>
- Tassopoulos, I. X., & Beligiannis, G. N. (30 de 11 de 2011). *WesternGreeceUniversityInstance5*. Obtido em 16 de 11 de 2016, de University of Twente: <https://www.utwente.nl/ctit/hstt/datasets/Greece/WesternGreeceUniversityInstance5/GreeceWesternGreeceUniversityInstance5.xml>
- University of Twente. (2009a). *Home*. Obtido em 02 de 12 de 2013, de University of Twente: <http://www.utwente.nl/>
- University of Twente. (2009b). *Benchmarking project for (High) School Timetabling*. Obtido em 12 de 11 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/>
- University of Twente. (2009c). *Australia > Background and literature*. Obtido em 30 de 12 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/datasets/Australia/Background/>

University of Twente. (2009d). *United Kingdom > Background and literature*. Obtido em 30 de 12 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/datasets/UnitedKingdom/Background/>

University of Twente. (2009e). *Finland > Background and literature*. Obtido em 30 de 12 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/datasets/Finland/Background/>

University of Twente. (2009f). *Greece > Background and literature*. Obtido em 30 de 12 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/datasets/Greece/Background/>

University of Twente. (2009g). *Netherlands > Background and literature*. Obtido em 30 de 12 de 2013, de University of Twente: <http://www.utwente.nl/ctit/hstt/datasets/Netherlands/Background/>

University of Twente. (07 de 2013h). *High School Timetable File Format Specification: Constraints*. Obtido em 30 de 12 de 2013, de University of Twente: <http://sydney.edu.au/engineering/it/~jeff/hseval.cgi?op=spec&part=constraints>

W3Schools. (1999). *JavaScript Tutorial*. Obtido em 10 de 01 de 2016, de W3Schools: <http://www.w3schools.com/js/default.asp>

World Wide Web Consortium (W3C). (29 de 10 de 2013). *Extensible Markup Language (XML)*. Obtido em 04 de 12 de 2013, de World Wide Web Consortium (W3C): <http://www.w3.org/XML/>

A. Apêndice

A.1 Exemplo de um formato XML para horários académicos da AFA

```
<HighSchoolTimetableArchive>
  <Instances>
    <Instance Id="3">
      <MetaData>
        <Name>AFA</Name>
        <Contributor>Academia da Força Aérea</Contributor>
        <Date>2015-01-14 14:58:19</Date>
        <Country>PORTUGAL</Country>
        <Description>Restrições para um horário</Description>
        <Remarks/>
      </MetaData>
      <Times>
        <TimeGroups>
          <Day Id="2">
            <Name>Segunda</Name>
          </Day>
          ...
        </TimeGroups>
        <TimeGroup Id="Mornings">
          <Name>Manhãs</Name>
        </TimeGroup>
      </TimeGroups>
      <Time Id="21">
        <Name>08:00</Name>
        <Day Reference="2"/>
        <TimeGroups>
          <TimeGroup Reference="All_Times"/>
          <TimeGroup Reference="First_Hours_Of_Days"/>
          <TimeGroup Reference="First_Hours_Of_Mornings"/>
          <TimeGroup Reference="Mornings"/>
        </TimeGroups>
      </Time>
      ...
    </Times>
  </Resources>
  <ResourceTypes>
    <ResourceType Id="Teacher">
      <Name>Teacher</Name>
    </ResourceType>
    ...
  </ResourceTypes>
  <ResourceGroup Id="AllTeachers">
```

```

    <Name>AllTeachers</Name>
    <ResourceType Reference="Teacher"/>
  </ResourceGroup>
</ResourceGroups>
  <ResourceGroup Id="AllClasses">
    <Name>AllClasses</Name>
    <ResourceType Reference="Class"/>
  </ResourceGroup>
...
</ResourceGroups>
<Resource Id="1T0016866">
  <Name>MANUEL M. DE</Name>
  <ResourceType Reference="Teacher"/>
  <ResourceGroups>
    <ResourceGroup Reference="AllTeachers"/>
  </ResourceGroups>
</Resource>
...
<Resource Id="3R35">
  <Name>PAV_DESP</Name>
  <ResourceType Reference="Room"/>
  <ResourceGroups>
    <ResourceGroup Reference="AllRooms"/>
  </ResourceGroups>
</Resource>
...
</Resources>
<Events>
  <EventGroups>
    <Course Id="11101">
      <Name>Mestrado em Aeronáutica Militar, Especialidade de Piloto Aviador</Name>
    </Course>
  ...
  <EventGroup Id="AllEvents">
    <Name>AllEvents</Name>
  </EventGroup>
</EventGroups>
  <Event Id="19">
    <Name>OTOPCM</Name>
    <Duration>1</Duration>
    <Course Reference="11101"/>
    <Resources>
      <Resource Reference="TFM_1">

```

```

    <Role>Teacher</Role>
    <ResourceType Reference="Teacher"/>
  </Resource>
  <Resource Reference="PILAV_1">
    <Role>Class</Role>
    <ResourceType Reference="Class"/>
  </Resource>
</Resources>
<EventGroups>
  <EventGroup Reference="All_Events"/>
</EventGroups>
</Event>
<Event Id="1397">
  <Name>EFI I</Name>
  <Duration>1</Duration>
  <Course Reference="11101"/>
...
</Event>
</Events>
<Constraints>
...
</Constraints>
</Instance>
</Instances>
<SolutionGroups>
  <SolutionGroup Id="AFA_2014-01-24">
    <MetaData>
      <Contributor>Academia da Força Aérea</Contributor>
      <Date>2015-01-14 14:58:19</Date>
      <Description>Solution</Description>
    </MetaData>
    <Solution Reference="3">
      <Events>
        <Event Reference="TEACH_1_PILAV_1_1">
          <Duration>1</Duration>
          <Time Reference="21" />
          <Resources />
        </Event>
      </Events>
    </SolutionGroup>
  </SolutionGroups>
...
  </Solution>
</SolutionGroup>
</SolutionGroups>

```

</HighSchoolTimetableArchive>

A.2 Exemplo num formato XML de restrições aos horários académicos da AFA

...

<Constraints>

...

<AssignTimeConstraint Id="AssignTimes_1">

<Name>AssignTimes</Name>

<Required>true</Required>

<Weight>1</Weight>

<CostFunction>Linear</CostFunction>

<AppliesTo>

<EventGroups>

<EventGroup Reference="AllEvents"/>

</EventGroups>

</AppliesTo>

</AssignTimeConstraint>

<PreferResourcesConstraint Id="PreferredResources_1">

<Name>PreferredResources</Name>

<Required>false</Required>

<Weight>2</Weight>

<CostFunction>Linear</CostFunction>

<AppliesTo>

<Events>

<Event Reference="1397"/>

<Event Reference="1405"/>

<Event Reference="1440"/>

<Event Reference="1454"/>

```

    </Events>

    </AppliesTo>

    <Resources>

        <Resource Reference="3R35"/>

    </Resources>

</PreferResourcesConstraint>

...

<PreferTimesConstraint Id="PreferredTimes_1">

    <Name>PreferredTimes</Name>

    <Required>true</Required>

    <Weight>1</Weight>

    <CostFunction>Linear</CostFunction>

    <AppliesTo>

        <Events>

            <Event Reference="1397"/>

            <Event Reference="1405"/>

            <Event Reference="1440"/>

            <Event Reference="1454"/>

        </Events>

    </AppliesTo>

    <TimeGroups>

        <TimeGroup Reference="Last_Hour_Of_Mornings"/>

        <TimeGroup Reference="Last_Hour_Of_Afternoons"/>

    </TimeGroups>

</PreferTimesConstraint>

...

<SpreadEventsConstraint Id="SpreadEvents_1">

    <Name>SpreadEvents</Name>

```

```

<Required>true</Required>

<Weight>1</Weight>

<CostFunction>Linear</CostFunction>

<AppliesTo>

  <EventGroups>

    <EventGroup Reference="AllEvents"/>

  </EventGroups>

</AppliesTo>

<TimeGroups>

  <TimeGroup Reference="2">

    <Minimum>0</Minimum>

    <Maximum>1</Maximum>

  </TimeGroup>

  <TimeGroup Reference="3">

    <Minimum>0</Minimum>

    <Maximum>1</Maximum>

  </TimeGroup>

  <TimeGroup Reference="4">

    <Minimum>0</Minimum>

    <Maximum>1</Maximum>

  </TimeGroup>

  <TimeGroup Reference="5">

    <Minimum>0</Minimum>

    <Maximum>1</Maximum>

  </TimeGroup>

  <TimeGroup Reference="6">

    <Minimum>0</Minimum>

    <Maximum>1</Maximum>

```



```

    </TimeGroup>

  </TimeGroups>

</SpreadEventsConstraint>

<AvoidClashesConstraint Id="No_Resource_Clashes_Constraint">

  <Name>No_Resource_Clashes_Constraint</Name>

  <Required>true</Required>

  <Weight>1</Weight>

  <CostFunction>Linear</CostFunction>

  <AppliesTo>

    <ResourceGroups>

      <ResourceGroup Reference="AllClasses"/>

      <ResourceGroup Reference="AllTeachers"/>

    </ResourceGroups>

  </AppliesTo>

</AvoidClashesConstraint>

<AvoidUnavailableTimesConstraint Id="AvoidUnavailableTimes_1">

  <Name>AvoidUnavailableTimes</Name>

  <Required>false</Required>

  <Weight>2</Weight>

  <CostFunction>Linear</CostFunction>

  <AppliesTo>

    <Resources>

      <Resource Reference="3R35"/>

    </Resources>

  </AppliesTo>

  <TimeGroups>

    <TimeGroup Reference="First_Hours_Of_Mornings"/>

    <TimeGroup Reference="First_Hours_Of_Afternoons"/>

```

```

    </TimeGroups>

</AvoidUnavailableTimesConstraint>

<LimitIdleTimesConstraint Id="LimitIdleTimes_1">

    <Name>LimitIdleTimes</Name>

    <Required>true</Required>

    <Weight>5</Weight>

    <CostFunction>SumSquares</CostFunction>

    <AppliesTo>

        <ResourceGroups>

            <ResourceGroup Reference="AllStudents"/>

        </ResourceGroups>

    </AppliesTo>

    <TimeGroups>

        <TimeGroup Reference="AllTimes"/>

    </TimeGroups>

    <Minimum>0</Minimum>

    <Maximum>0</Maximum>

</LimitIdleTimesConstraint>

</Constraints>

...

```

A.3 Resultados experimentais baseados no horário da AFA com a alteração da tarde livre para outro dia

RRHC	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	5621	00:05,621	-1422,4	00:01,422	0	0
2	30000000	30	5721	00:05,721	-1322,4	00:01,322	0	0
3	30000000	30	8081	00:08,081	1037,6	00:01,038	0	0
4	30000000	30	7904	00:07,904	860,6	00:00,861	0	0
5	30000000	30	7890	00:07,890	846,6	00:00,847	0	0
Média			7043,4	00:07,043			0	0
Desvio Padrão			1255,58	00:01,256			0,00	0,00

LAHC	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	8091	00:08,091	-81,6	00:00,082	0	0
2	30000000	30	8319	00:08,319	146,4	00:00,146	0	0
3	30000000	30	8087	00:08,087	-85,6	00:00,086	0	0
4	30000000	30	8205	00:08,205	32,4	00:00,032	0	0
5	30000000	30	8161	00:08,161	-11,6	00:00,012	0	0
Média			8172,6	00:08,173			0	0
Desvio Padrão			95,66	00:00,096			0,00	0,00

SCHC	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	8236	00:08,236	-6,4	00:00,006	0	0
2	30000000	30	8052	00:08,052	-190,4	00:00,190	0	0
3	30000000	30	8214	00:08,214	-28,4	00:00,028	0	0
4	30000000	30	8353	00:08,353	110,6	00:00,111	0	0
5	30000000	30	8357	00:08,357	114,6	00:00,115	0	0
Média			8242,4	00:08,242			0	0
Desvio Padrão			124,97	00:00,125			0,00	0,00

SA	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	104486	01:44,486	3134,8	00:03,135	3000004	7
2	30000000	30	106197	01:46,197	4845,8	00:04,846	4000003	7
3	30000000	30	99110	01:39,110	-2241,2	00:02,241	3000002	5
4	30000000	30	93484	01:33,484	-7867,2	00:07,867	2000000	2
5	30000000	30	103479	01:43,479	2127,8	00:02,128	4000001	5
Média			101351,2	01:41,351			3200002	5,2
Desvio Padrão			5117,42	00:05,117			836660,62	2,05

TS	Valores Iniciais		Tempo de Execução		Desvio		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	30000000	30	8270	00:08,270	93,2	00:00,093	0	0
2	30000000	30	8162	00:08,162	-14,8	00:00,015	0	0
3	30000000	30	8126	00:08,126	-50,8	00:00,051	0	0
4	30000000	30	8146	00:08,146	-30,8	00:00,031	0	0
5	30000000	30	8180	00:08,180	3,2	00:00,003	0	0
Média			8176,8	00:08,177			0	0
Desvio Padrão			55,78	00:00,056			0,00	0,00

Médias	Tempo de Execução		Valores Finais	
Técnicas	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	7043,4	00:07,043	0	0
LAHC	8172,6	00:08,173	0	0
SCHC	8242,4	00:08,242	0	0
SA	101351,2	01:41,351	3200002	5,2
TS	8176,8	00:08,177	0	0

A.4 Resultados experimentais baseados no horário da AFA com a indisponibilidade de dois professores

RRHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	10000000	10	9205	00:09,205	19	00:00,019	0	0
2	10000000	10	9173	00:09,173	-13	00:00,013	0	0
3	10000000	10	9137	00:09,137	-49	00:00,049	0	0
4	10000000	10	9225	00:09,225	39	00:00,039	0	0
5	10000000	10	9190	00:09,190	4	00:00,004	0	0
Média			9186	00:09,186			0	0
Desvio Padrão			33,42	00:00,033			0,00	0,00

LAHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	10000000	10	9288	00:09,288	-24	00:00,024	0	0
2	10000000	10	9228	00:09,228	-84	00:00,084	0	0
3	10000000	10	9224	00:09,224	-88	00:00,088	0	0
4	10000000	10	9444	00:09,444	132	00:00,132	0	0
5	10000000	10	9376	00:09,376	64	00:00,064	0	0
Média			9312	00:09,312			0	0
Desvio Padrão			96,04	00:00,096			0,00	0,00

SCHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	10000000	10	9043	00:09,043	-26,2	00:00,026	0	0
2	10000000	10	9105	00:09,105	35,8	00:00,036	0	0
3	10000000	10	9017	00:09,017	-52,2	00:00,052	0	0
4	10000000	10	9109	00:09,109	39,8	00:00,040	0	0
5	10000000	10	9072	00:09,072	2,8	00:00,003	0	0
Média			9069,2	00:09,069			0	0
Desvio Padrão			39,64	00:00,040			0,00	0,00

SA	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	10000000	10	6878	00:06,878	-20671,4	00:20,671	1	1
2	10000000	10	36498	00:36,498	8948,6	00:08,949	1	1
3	10000000	10	60711	01:00,711	33161,6	00:33,162	0	0
4	10000000	10	10012	00:10,012	-17537,4	00:17,537	1	1
5	10000000	10	23648	00:23,648	-3901,4	00:03,901	1000001	2
Média			27549,4	00:27,549			200000,8	1
Desvio Padrão			21965,07	00:21,965			447213,71	0,71

TS	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	10000000	10	8951	00:08,951	74	00:00,074	0	0
2	10000000	10	8759	00:08,759	-118	00:00,118	0	0
3	10000000	10	8866	00:08,866	-11	00:00,011	0	0
4	10000000	10	8925	00:08,925	48	00:00,048	0	0
5	10000000	10	8884	00:08,884	7	00:00,007	0	0
Média			8877	00:08,877			0	0
Desvio Padrão			73,95	00:00,074			0,00	0,00

Médias	Tempo de Execução		Valores Finais	
Técnicas	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	9186	00:09,186	0	0
LAHC	9312	00:09,312	0	0
SCHC	9069,2	00:09,069	0	0
SA	27549,4	00:27,549	200000,8	1
TS	8877	00:08,877	0	0

A.5 Resultados experimentais baseados no horário da AFA com a indisponibilidade de uma sala

RRHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	18000000	18	47747	00:47,747	-23429,8	00:23,430	0	0
2	18000000	18	79765	01:19,765	8588,2	00:08,588	0	0
3	18000000	18	77525	01:17,525	6348,2	00:06,348	0	0
4	18000000	18	75023	01:15,023	3846,2	00:03,846	0	0
5	18000000	18	75824	01:15,824	4647,2	00:04,647	0	0
Média			71176,8	01:11,177			0	0
Desvio Padrão			13223,11	00:13,223			0,00	0,00

LAHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	18000000	18	88795	01:28,795	-998,2	00:00,998	0	0
2	18000000	18	91575	01:31,575	1781,8	00:01,782	0	0
3	18000000	18	91897	01:31,897	2103,8	00:02,104	0	0
4	18000000	18	91624	01:31,624	1830,8	00:01,831	0	0
5	18000000	18	85075	01:25,075	-4718,2	00:04,718	0	0
Média			89793,2	01:29,793			0	0
Desvio Padrão			2924,48	00:02,924			0,00	0,00

SCHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	18000000	18	79099	01:19,099	1809,6	00:01,810	0	0
2	18000000	18	76262	01:16,262	-1027,4	00:01,027	0	0
3	18000000	18	77740	01:17,740	450,6	00:00,451	0	0
4	18000000	18	75615	01:15,615	-1674,4	00:01,674	0	0
5	18000000	18	77731	01:17,731	441,6	00:00,442	0	0
Média			77289,4	01:17,289			0	0
Desvio Padrão			1372,21	00:01,372			0,00	0,00

SA	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	18000000	18	113324	01:53,324	2175,4	00:02,175	9000000	9
2	18000000	18	110193	01:50,193	-955,6	00:00,956	9000000	9
3	18000000	18	112211	01:52,211	1062,4	00:01,062	9000002	11
4	18000000	18	110262	01:50,262	-886,6	00:00,887	9000000	9
5	18000000	18	109753	01:49,753	-1395,6	00:01,396	9000000	9
Média			111148,6	01:51,149			9000000,4	9,4
Desvio Padrão			1541,75	00:01,542			0,89	0,89

TB	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	18000000	18	76313	01:16,313	1712,4	00:01,712	0	0
2	18000000	18	74257	01:14,257	-343,6	00:00,344	0	0
3	18000000	18	73279	01:13,279	-1321,6	00:01,322	0	0
4	18000000	18	73468	01:13,468	-1132,6	00:01,133	0	0
5	18000000	18	75686	01:15,686	1085,4	00:01,085	0	0
Média			74600,6	01:14,601			0	0
Desvio Padrão			1347,02	00:01,347			0,00	0,00

Médias	Tempo de Execução		Valores Finais	
Técnicas	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	71176,8	01:11,177	0	0
LAHC	89793,2	01:29,793	0	0
SCHC	77289,4	01:17,289	0	0
SA	111148,6	01:51,149	9000000,4	9,4
TS	74600,6	01:14,601	0	0

A.6 Resultados experimentais através de geração aleatória de horários para a AFA

RRHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	232000680	282	50726	00:50,726	-21498,6	00:21,499	0	0
2	252000856	298	163873	02:43,873	91648,4	01:31,648	0	0
3	248000618	298	61357	01:01,357	-10867,6	00:10,868	0	0
4	262000436	286	49121	00:49,121	-23103,6	00:23,104	0	0
5	222000682	280	57066	00:57,066	-15158,6	00:15,159	0	0
6	230000492	254	78533	01:18,533	6308,4	00:06,308	0	0
7	240000738	298	67241	01:07,241	-4983,6	00:04,984	0	0
8	240000682	276	91740	01:31,740	19515,4	00:19,515	0	0
9	228000500	258	43159	00:43,159	-29065,6	00:29,066	0	0
10	226000496	284	59430	00:59,430	-12794,6	00:12,795	0	0
Média			72224,6	01:12,225			0	0
Desvio Padrão			35272,40	00:35,272			0,00	0,00

LAHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	230000794	298	64146	01:04,146	2224,5	00:02,225	0	0
2	238000616	298	47500	00:47,500	-14421,5	00:14,422	0	0
3	230000976	298	96227	01:36,227	34305,5	00:34,305	0	0
4	244000976	298	92198	01:32,198	30276,5	00:30,276	0	0
5	246000498	280	52611	00:52,611	-9310,5	00:09,311	0	0
6	240000434	256	35411	00:35,411	-26510,5	00:26,510	0	0
7	210000380	258	35310	00:35,310	-26611,5	00:26,612	0	0
8	208000680	298	51271	00:51,271	-10650,5	00:10,651	0	0
9	258000854	298	86906	01:26,906	24984,5	00:24,984	0	0
10	216000670	298	57635	00:57,635	-4286,5	00:04,287	0	0
Média			61921,5	01:01,922			0	0
Desvio Padrão			22509,65	00:22,510			0,00	0,00

SCHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	254000494	284	79863	01:19,863	-403424,1	06:43,424	0	0
2	234000678	286	47930	00:47,930	-435357,1	07:15,357	0	0
3	230000676	298	2159085	35:59,085	1675797,9	27:55,798	1000001	2
4	238000612	298	1539953	25:39,953	1056665,9	17:36,666	1000000	1
5	200000742	298	38803	00:38,803	-444484,1	07:24,484	0	0
6	228000500	282	167485	02:47,485	-315802,1	05:15,802	0	0
7	206000620	298	56788	00:56,788	-426499,1	07:06,499	0	0
8	220000608	282	291213	04:51,213	-192074,1	03:12,074	0	0
9	236000736	298	173725	02:53,725	-309562,1	05:09,562	0	0
10	246000672	284	278026	04:38,026	-205261,1	03:25,261	0	0
Média			483287,1	08:03,287			200000,1	0,3
Desvio Padrão			740208,68	12:20,209			421637,23	0,67

SA	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	218000560	282	163396	02:43,396	6735,2	00:06,735	1000001	2
2	237000377	274	139492	02:19,492	-17168,8	00:17,169	3	3
3	216000505	271	154953	02:34,953	-1707,8	00:01,708	2000000	2
4	214000556	300	165393	02:45,393	8732,2	00:08,732	1	1
5	218000559	299	146779	02:26,779	-9881,8	00:09,882	1000000	1
6	225000612	285	170838	02:50,838	14177,2	00:14,177	1000000	1
7	221000498	255	155842	02:35,842	-818,8	00:00,819	2000002	4
8	232000734	280	162871	02:42,871	6210,2	00:06,210	5000002	7
9	215000676	305	146573	02:26,573	-10087,8	00:10,088	7000023	22
10	255000580	305	160471	02:40,471	3810,2	00:03,810	3000001	4
Média			156660,8	02:36,661			2200003,3	4,7
Desvio Padrão			9851,45	00:09,851			2250930,99	6,36

TB	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	270000730	276	88250	01:28,250	33004,2	00:33,004	0	0
2	224000496	298	43985	00:43,985	-11260,8	00:11,261	0	0
3	234000790	298	63708	01:03,708	8462,2	00:08,462	0	0
4	226000492	262	55086	00:55,086	-159,8	00:00,160	0	0
5	244000320	266	48183	00:48,183	-7062,8	00:07,063	0	0
6	244000496	284	44858	00:44,858	-10387,8	00:10,388	0	0
7	238000492	280	44017	00:44,017	-11228,8	00:11,229	0	0
8	208000740	298	43927	00:43,927	-11318,8	00:11,319	0	0
9	212000792	298	61963	01:01,963	6717,2	00:06,717	0	0
10	244000726	258	58481	00:58,481	3235,2	00:03,235	0	0
Média			55245,8	00:55,246			0	0
Desvio Padrão			13965,41	00:13,965			0,00	0,00

Médias	Tempo de Execução		Valores Finais	
Técnicas	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	72224,6	01:12,225	0	0
LAHC	61921,5	01:01,922	0	0
SCHC	483287,1	08:03,287	200000,1	0,3
SA	156660,8	02:36,661	2200003,3	4,7
TS	55245,8	00:55,246	0	0

A.7 Resultados experimentais através de geração aleatória de horários para escola da Grécia

RRHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	2390000001	240	179765	02:59,765	-27419,6	00:27,420	0	0
2	2400000000	240	237433	03:57,433	30248,4	00:30,248	0	0
3	2440000000	244	201569	03:21,569	-5615,6	00:05,616	0	0
4	2520000000	252	199708	03:19,708	-7476,6	00:07,477	0	0
5	2600000000	260	217448	03:37,448	10263,4	00:10,263	0	0
Média			207184,6	03:27,185			0	0
Desvio Padrão			21561,38	00:21,561			0,00	0,00

LAHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	2400000000	240	195464	03:15,464	3985,8	00:03,986	0	0
2	2540000000	254	207751	03:27,751	16272,8	00:16,273	0	0
3	2340000000	234	183397	03:03,397	-8081,2	00:08,081	0	0
4	2420000000	242	179661	02:59,661	-11817,2	00:11,817	0	0
5	2380000000	238	191118	03:11,118	-360,2	00:00,360	0	0
Média			191478,2	03:11,478			0	0
Desvio Padrão			11020,12	00:11,020			0,00	0,00

SCHC	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	2600000000	260	200150	03:20,150	-3431,6	00:03,432	0	0
2	2320000000	232	214784	03:34,784	11202,4	00:11,202	0	0
3	2500000000	250	204725	03:24,725	1143,4	00:01,143	0	0
4	2460000000	246	196854	03:16,854	-6727,6	00:06,728	0	0
5	2480000000	248	201395	03:21,395	-2186,6	00:02,187	0	0
Média			203581,6	03:23,582			0	0
Desvio Padrão			6866,93	00:06,867			0,00	0,00

SA	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	2540000000	254	741364	12:21,364	-12146	00:12,146	27000000	27
2	3260000000	326	759060	12:39,060	5550	00:05,550	31000000	31
3	2640000000	264	743039	12:23,039	-10471	00:10,471	22000000	22
4	2850000000	285	772176	12:52,176	18666	00:18,666	37000000	37
5	2830000000	283	751911	12:31,911	-1599	00:01,599	36000000	36
Média			753510	12:33,510			30600000	30,6
Desvio Padrão			12638,69	00:12,639			6268971,21	6,27

TB	Valores Iniciais		Tempo de Execução		Desvio Padrão		Valores Finais	
Testes	Custo	Violações	Milisegundos	mm:ss,SSS	Milisegundos	mm:ss,SSS	Custo	Violações
1	254000000	254	196897	03:16,897	9932,4	00:09,932	0	0
2	246000000	246	203284	03:23,284	16319,4	00:16,319	0	0
3	238000000	238	170163	02:50,163	-16801,6	00:16,802	0	0
4	254000000	254	189229	03:09,229	2264,4	00:02,264	0	0
5	224000000	224	175250	02:55,250	-11714,6	00:11,715	0	0
Média			186964,6	03:06,965			0	0
Desvio Padrão			14050,16	00:14,050			0,00	0,00

Médias	Tempo de Execução		Valores Finais	
Técnicas	Milisegundos	mm:ss,SSS	Custo	Violações
RRHC	207184,6	03:27,185	0	0
LAHC	191478,2	03:11,478	0	0
SCHC	203581,6	03:23,582	0	0
SA	753510	12:33,510	30600000	30,6
TS	186964,6	03:06,965	0	0